SUMMER 1984

An Argus Specialist Publication

Personal SOFTWARE

£2.10

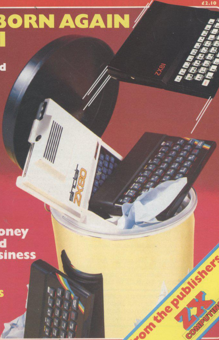
THE BORN AGAIN ZX81

Games and teasers to drive you wild

Get educated and have fun

Make money and mind your business

Graphics come to life



From the publishers of ZX COMPUTING

A NEW, IMPROVED ZX81 KEYBOARD AT THE SAME OLD PRICE. £9.95.

There's only one thing wrong with the ZX81. Its keyboard.

Or rather its lack of one.

Since it's flat your fingers don't feel as if there's any response to the pressure put on the keys.

In other words, you're not quite sure which keys you've pressed until the screen actually tells you.

Our new, improved push button keyboard changes all that.

It matches the ZX81 perfectly. And the keys give a real calculator-type feel.

To set it up all you have to do is peel off the adhesive backing and stick it on top of the ZX81 touchpad.

Because no tampering or soldering is involved the guarantee is not affected. And it will last for up to 3½ million operations.

But our keyboard doesn't just come loaded with features. With it comes a separate overlay and a set of coloured stick-on labels to make game playing easier.

It's yours for the original price of £9.95.

Whichever way you look at it, we think you'll agree that it's a keyboard that's quite outstanding.

Orders to Filesixty Ltd., FREEPOST, London W9 2BR.
Cheques/PO made payable to Filesixty Ltd.
Please send me........(qty) Keyboards at £9.95 each (including VAT and P&P).
Total £
Name BLOCK CAPITALS
Address

FILESIXTY

Filesixty Ltd., 25 Chippenham Mews, London W9 2AN, England. Tel: 01-289 3059. Telex: 268 048 EXTLDN G 4087.

SUMMER 1984
Personal
SOFTWARE
THE BORN AGAIN ZX81
Games and teasers to drive you wild
Get educated and have fun
Make money and mind your business
Graphics come to life
From the publishers of ZX Computing

Volume 3
No 1
Summer 1984

Editor: Wendy J Palmer
Sales Executive:
Penny Scoular
Advertisement Manager:
Barry Bingham
Advertisement Copy Control:
Ann McDermott
Managing Editor:
Ron Harris
Chief Executive: T J Connell
Origination and design by MM Design & Print

Personal Software is a quarterly magazine appearing on the third Friday in May, August, November and February.

Distribution by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS Tel: 01-247 8233. Printing by: Alabaster Passmore & Sons Ltd, Tovil, Maidstone, Kent.

The contents of this publication including all articles, designs, plans, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights conferred by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the prior written consent of the Company. © 1984 Argus Specialist Publications Limited.

Subscription Rates : UK £8.40 for four issues including postage. Airmail and other rates upon application to Personal Software Subscriptions, Infonet Ltd, Times House, 179 The Marlowes, Hemel Hempstead, Herts HP1 1BB.

CONTENTS

Editorial & Advertisement Office
No 1 Golden Square, London W1R 3AB
Telephone: 01-437 0626 Telex: 8811896

Preface 4

Invincible 5
Try playing hard to get — the aliens are out to get you!

Motor Madness 6
Enter the Grand Prix and drive yourself mad.

Caterpillar 2 8
Watch out you don't become addicted!

Crazy Ballooning 10
Fly high and fast and watch out for low-flying aircraft.

Talking Turkey 12
Let your micro create a new dictionary.

Worms 13
Surround yourself in machine code.

Backgammon 16
See if you can bear off before the computer.

3D Mover 22
Create three-dimensional objects in 16K.

Just another Brick 24
Build yourself up to playing games.

Number Tumbler 27
Is your number up? Read on and find out.

3D Cubes 28
Three-dimensional representations of cubes on the ZX81.

Grand Prix Driver 31
Strap yourself in and prepare for the race of your life.

Equation Solver 36
Solve equations using the bisection method.

Minefield 39
You really must watch your step.

Death Race 41
Drive your ZX81 to the limit!

Towers of Hanoi 42
The ZX81 solution to this age-old puzzle.

Sheepdog Trial 44
Head 'em up and move 'em out!

User Character Set 46
All sorts of characters on the ZX81.

The Sharp End 48
Three great little games to get you going!

Muncher II 50
Try to escape from the monsters.

The Right Chord 53
Tricky guitar chords made easy.

Enemy Satellite 55
Home in on the enemy.

On Target 56
Two great programs — Target and Missile Launcher.

Smaug's Lair 58
Can you destroy the dragon's lair?

Naval Manoeuvres 62
Get your fleet in action.

House of Horror 66
I hope you're not faint-hearted

Drop Out 68
A ZX81 version of the old arcade game.

Maths Maze 70
A great game with an educational twist.

Naught But A Game 73
Don't get cross at the noughts.

Peeper 76
A paranoid program, so watch out!

Five Card Trick 78
A fabulous implementation of the card game, Pontoon.

Profit Prophet 81
Serious stuff to help you get to grips with your business.

Stock Control 85
Keep efficient control of your stock.

Air Raid 88
Run for cover and have a smashing time!

Coding Graphics 92
Increase your character size with this utility.

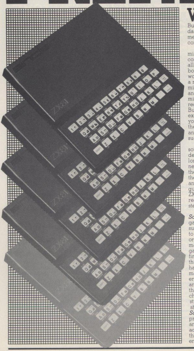
Spiralling into Graphics 96
Become artistic with a spirograph program.

Shooter 98
Simple but fun to play.

Credits: Our grateful thanks to the following individuals for their assistance in the production of this issue. Mark, Gary, Bill, Bill, Nigel, Tom, William, Adrian, Sam, Fiona and others almost too numerous to mention.

Note: Sinclair, ZX, ZX Spectrum, ZX Microdrive, ZX Interface are all registered trademarks of Sinclair Research Ltd.

PREFACE



What's this, I hear someone ask — a magazine for the ZX81? But surely the ZX81 has had its day and is now no more than a memory? Well, if you are sitting comfortably I'll try to explain. . .

At a rough estimate, there a million ZX81s out there in computer-land, and even allowing for those people who bought one as a 'one-day wonder' or have 'progressed' to a more sophisticated microcomputer, there must be an awfully large number of micro owners out there who regularly plug in their ZX81. But have you tried to find some exciting new software lately in your local high street store for the ZX81? I'd like to hear from anybody that has had recent success in this venture!

It would appear that the software manufacturers have decided that the ZX81 is no longer a viable launchpad for new programs and now devote their time, money and effort to the more profitable machines — and who suffers most? You've guessed it — the multitude of ZX81 owners. And there, dear reader, is where we have stepped in.

This issue of ***Personal Software*** is intended to fill the gap left by the commercial suppliers. Most of the programs to be found in these pages were originally printed in our sister magazine, ***ZX Computing***. The game of Backgammon appeared first in ***Computing Today***. Since they first appeared in print, we have carefully checked them to make sure they contain no errors, and so if you experience any difficulty in getting any of the programs to work, please check them again. If you are still having problems you should write to ***Personal Software*** stating clearly what problem you have encountered and enclosing a stamped self-addressed envelope. We regret that we cannot answer telephone enquiries.

INVINCIBLE

Try playing hard to get in this program — the aliens are out to get you.

The object of this game is to shoot the alien spaceship five times before it gets to shoot you. You use the 'I' key and the 'Q' key to move left and right respectively. You can fire at the spaceship using the 'O' key.

You must make sure your shots are aimed at the black squares of the spaceship — all other hits are counted as naught. The winner of the game is the first one to get five hits. The seemingly random firing of the spaceship is accomplished using the RND function.

The game is great fun to play and beating the spaceship is no easy task!

```
   1 PRINT "          ██████████
█","         █INVINCIBLE█","
 ████████████"
   2 PRINT AT 5,0;"USE 1 TO MOVE
 LEFT","USE Q T0 MOVE RIGHT","US
E O TO FIRE"
   3 PRINT AT 10.0;"YOUR SHOTS C
AN PENETRATE THE ","BLACK SQUARE
S ONLY"
   4 PRINT AT 14,0."FIRST ONE TO
 SCORE 5 HITS WINS"
  10 PRINT
  12 PRINT "PRESS ANY KEY TO STA
RT"
  20 PAUSE 4E4
  90 CLS
 100 LET A=0
 101 LET SH=0
 102 LET ZX=0
 103 LET SK=0
 104 LET ZK=0
 105 LET A$="█▒█▒█▒█▒█▒█▒█▒█▒█▒█▒
▒█▒█▒█▒█▒█▒"
 110 LET B=1
 120 LET C=2
 130 LET D=3
 140 LET E=16
 150 LET H=INT (RND*25)
 160 LET F=17
 170 LET G=INT (RND*25)
 300 LET Q=INT (RND*4)
 500 PRINT AT A,A;"██████████
████████████████";AT E,A;"█
 510 PRINT AT B,H;" ▛▜ "
 520 PRINT AT F,G;" ▙▟ "
 530 PRINT AT 9,A;A$;
 540 IF Q=1 THEN LET H=H-1
 550 IF Q=2 THEN LET H=H+1
 555 IF Q=3 THEN GOSUB 1500
 560 IF H>=26 THEN LET H=26
 561 IF H<=0 THEN LET H=0
 565 IF G>=26 THEN LET G=26
 566 IF G<=0 THEN LET G=0
 600 IF INKEY$="1" THEN LET G=G-
1
 610 IF INKEY$="Q" THEN LET G=G+
1
 620 IF INKEY$="0" THEN GOSUB 10
00
 900 GOTO 300
1005 LET SH=SH+1
1006 FOR I=14 TO 1 STEP -1
1010 PRINT AT I,G+2;".";
1020 PRINT AT I,G+2;" ";
1030 IF PEEK (PEEK 16398+256*PEE
K 16399)=128 THEN GOTO 2000
1040 IF PEEK (PEEK 16398+256+PEE
K 16399)=129 THEN GOTO 1200
1060 NEXT I
1100 RETURN
1200 PRINT AT B,H;"*****"
1205 LET SK=SK+1
1210 IF SK=5 THEN GOTO 2500
1220 GOTO 300
1500 FOR I=2 TO 17
1501 PRINT AT I,H+2;"."
1505 PRINT AT I,H+2;" "
1510 NEXT I
1512 LET ZX=ZX+1
1520 IF H+2=G+2 THEN GOTO 1700
1600 RETURN
1700 PRINT AT F,G;"+++++"
1705 LET ZK=ZK+1
1710 IF ZK=5 THEN GOTO 2500
1720 GOTO 300
2000 PRINT "▒"
2002 GOTO 300
2500 CLS
2510 PRINT AT 1,0;"**********GAM
E OVER**********"
2520 PRINT AT 3,0;"ZX81 SHOTS =
";ZX
2530 PRINT
2540 PRINT "ZX81 KILLS = ";ZK
2550 PRINT
2560 PRINT
2570 PRINT "HUMAN SHOTS = ";SH
2580 PRINT
2590 PRINT "HUMAN KILLS = ";SK
2600 PRINT
2610 PRINT
2620 PRINT "PRESS ANY KEY TO TRY
 AGAIN"
2630 PAUSE 4E4
2660 GOTO 90
```

Keith Ditchburn

MOTOR MADNESS

Enter the Grand Prix and drive yourself mad.

You are a racing driver in the Grand Prix and you must steer your car using the '5' and '8' keys to move left and right respectively.

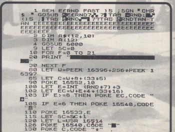
The machine code takes up to 50 bytes and should be located in a REM statement with the starting address 16514. The routine you'll need to POKE this into your computer follows below:

```
1  REM 50 Es
10 LET A = 16514
20 INPUT N
30 PRINT N;",",
35 POKE A,N
40 LET A = A + 1
50 GOTO 20
```

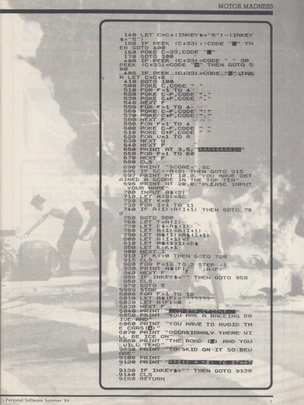
Once this program has been entered into the computer you should type in the following numbers. (The commas separating the numbers represent Newline.)

Okay, you're on your own now — happy racing, and try and keep out of the fast lane!

```
42, 12,64,229, 17, 33, 0, 25, 209, 1, 214, 2 237, 176, 42, 12, 64,
35, 6, 15, 35, 5, 194, 150, 64, 14, 126, 6, 16, 17, 33, 0, 25, 5,
194, 162, 64, 6, 5, 5, 35, 113, 194, 169, 64, 201,
```



```
   1 REM E£RND FAST )5 ;SGN "CHR
$ " GOSUB ▛E£RND7▞ 7▌TAB ▀RND:▛▞
()5 ;▌TAB ▙RND▛▀▌ 7?TAB ▙RNDTAN T
AN EEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEE
   2 DIM A$(12,10)
   3 DIM A(12)
   4 GOSUB 6000
   9 LET SC=0
  10 FOR F=0 TO 21
  20 PRINT "██████████████████████████
██"
  30 NEXT F
  80 LET W=PEEK 16396+256*PEEK 1
6397
  85 LET C=W+8+(33*5)
  90 POKE 16552,10
 100 LET E=INT (RND*7)+3
 102 LET EC=W+E+4+(33*16)
 103 IF E>=8 THEN POKE EC,CODE "
▣"
 105 IF E=6 THEN POKE 16540,CODE
"▒"
 110 POKE 16533,E
 115 LET SC=SC+1
 120 LET L=USR 16514
 126 POKE 16540,CODE "█"
 130 POKE C,CODE "*"
```



```
 140 LET C=C+(INKEY$="8")-(INKEY
$="5")
 150 IF PEEK (C+33)<>CODE "█" TH
EN GOTO 400
 160 POKE C-33,CODE "█"
 170 GOTO 100
 400 IF PEEK (C+33)=CODE " " OR
PEEK (C+33)=CODE "▣" THEN GOTO 5
00
 405 IF PEEK (C+33)=CODE "▒" THE
N LET C=C+2
 410 GOTO 100
 500 POKE C,CODE " "
 510 FOR F=1 TO 4
 520 POKE C-F,CODE "."
 530 POKE C+F,CODE "."
 540 NEXT F
 550 FOR F=1 TO 4
 560 POKE C-F,CODE ":"
 570 POKE C+F,CODE ":"
 580 NEXT F
 590 FOR F=1 TO 4
 600 POKE C-F,CODE " "
 610 POKE C+F,CODE " "
 620 FOR U=1 TO 8
 630 NEXT U
 640 NEXT F
 650 PRINT AT 3,5;"[illegible]"
 660 FOR F=1 TO 60
 670 NEXT F
 680 CLS
 690 PRINT "SCORE=";SC
 695 IF SC<=A(2) THEN GOTO 915
 697 PRINT AT 10,0;"YOU HAVE OBT
AINED A SCORE IN THE TOP TEN"
 698 PRINT AT 20,0;"PLEASE INPUT
 YOUR NAME"
 700 INPUT A$(2)
 710 LET A(2)=SC
 720 LET K=0
 730 FOR I=1 TO 11
 740 IF A(I)>A(I+1) THEN GOTO 76
0
 750 GOTO 900
 760 LET T=A(I)
 770 LET D$=A$(I)
 780 LET A(I)=A(I+1)
 790 LET A$(I)=A$(I+1)
 800 LET A(I+1)=T
 810 LET A$(I+1)=D$
 850 LET K=K+1
 900 NEXT I
 910 IF K<>0 THEN GOTO 720
 915 CLS
 920 FOR F=12 TO 2 STEP -1
 930 PRINT A$(F);" ";A(F)
 940 NEXT F
 950 IF INKEY$="" THEN GOTO 950
 960 CLS
 970 GOTO 5
5999 STOP
6000 FOR F=1 TO 12
6010 LET A$(F)="??????"
6020 LET A(F)=0
6030 NEXT F
6040 PRINT "INSTRUCTIONS."
6050 PRINT "YOU ARE A RACEING DR
IVE AND"
6060 PRINT "YOU HAVE TO AVOID TH
E CARS(▣)"
6070 PRINT "OCCASIONALY THERE WI
LL BE ICE ON"
6080 PRINT "THE ROAD (▒) AND YOU
 WILL TEND"
9090 PRINT "TO SKID ON IT SO BEW
ARE"
9100 PRINT
9120 PRINT "PRESS A KEY TO START
"
9130 IF INKEY$="" THEN GOTO 9130
9140 CLS
9150 RETURN
```

Derek Scott

CATERPILLAR 2

Watch out you don't become addicted.

You'll need some skillful playing if you reckon on getting out of the field in one piece in this program.

The object of the game is to guide a caterpillar through a field eating cabbage leaves (made up of 'fuzzy' squares). The caterpillar must avoid touching the walls (made up of inverse spaces) and eventually escape through a small gap in the wall.

The game has a high score facility, which asks for your name if you have beaten the previous player's high score. There are three skill levels, but you won't be able to get onto level three unless you have managed to score 1,000 points or more.

PROGRAM STRUCTURE

Lines	Description
10-115	The instructions and the skill level.
200-500	The main body of the program.
600-660	The high score facility.
1500-1530	The routine to decide the skill level.
2000-2010	The routine to decide whether a player is eligible to play on level three.

The listing for Caterpillar 2.

```
 10 SLOW
 15 LET H=0
 20 PRINT TAB 1;"CATERPILLAR 2
- D SCOTT, 1982"
 25 PRINT
 30 PRINT "DO YOU WANT A GAME?
        (YES/NO)"
 40 IF INKEY$="Y" THEN GOTO 70
 50 IF INKEY$<>"N" THEN GOTO 40
 60 PRINT "OK, GOODBYE..."
 65 STOP
 70 CLS
 72 PRINT "DO YOU WANT INSTRUCT
IONS?       (YES/NO)"
 75 LET S=0
 78 IF INKEY$<>"" THEN GOTO 78
 80 IF INKEY$="N" THEN GOTO 105
 85 IF INKEY$<>"Y" THEN GOTO 80
 90 CLS
 100 PRINT "CATERPILLAR 2 - INST
RUCTIONS","THIS IS ""CATERPILLAR
 2""-THATS 1 MORE THAN 1 AND 1 L
ESS THAN 3,  AT LEAST I THINK IT
 IS.","THE OBJECT OF THE GAME IS
 TO","GUIDE THE CATERPILLAR THRO
UGH","THE MAZE, USING KEYS ""5""
 AND",""8"". PICK UP THE CABBAG
E LEAVES AS YOU GO. AVOID ALL BL
ACK","OBSTACLES.",,"BONUS POINTS
 FOR LEAVES EATEN."
 101 PRINT AT 18,0;"PRESS NEWLIN
E TO CONTINUE"
 102 IF CODE INKEY$<>118 THEN GO
TO 102
 104 CLS
 105 PRINT "THERE ARE THREE SKIL
L LEVELS:","1 - BEGINNERS","2 -
AVERAGE","3 - MAGIC FINGERS","CH
OOSE YOUR LEVEL"
 110 INPUT Z
 111 IF Z<1 OR Z>3 THEN GOTO 110
 112 IF Z<>3 THEN GOTO 120
 115 GOTO 2000
 120 CLS
 200 FOR A=1 TO 21
 210 PRINT "■
        ■"
 220 NEXT A
 230 LET P=5
 250 LET A=INT (RND*25+1)
 260 GOSUB 400
 270 GOSUB 450
 280 IF R=128 THEN GOTO 600
 290 IF RND<.7 THEN GOTO 260
 300 FOR A=1 TO 21
 310 SCROLL
 320 PRINT AT 21,0;"■
        ■"
 330 IF RND<.3 AND A>1 THEN PRIN
T AT 21,P;"▒"
 340 GOSUB 450
 350 IF R=128 THEN/GOTO 600
 360 NEXT A
 370 GOTO 250
 400 SCROLL
 410 PRINT AT 21,0;"■■■■■■■■■■■■■
■■■■■■■■■■■■■■■■■"
 420 GOSUB 1500
 430 RETURN
 450 PRINT AT 6,P;
 460 LET R=PEEK (PEEK 16398+256*
PEEK 16399)
 470 PRINT "▒"
 480 LET S=S+1
 490 LET P=P+(INKEY$="8")-(INKEY
$="5")
 500 RETURN
 600 CLS
 601 IF H<S THEN PRINT "NEW HIGH
 SCORE - ENTER NAME"
 602 IF H>=S THEN GOTO 610
 603 INPUT O$
 605 IF H<S THEN LET H=S
 610 CLS
 620 PRINT AT 0,6;"HIGH SCORE=";
H;AT 1,8;"BY ";O$
 625 PRINT AT 4,10;"SCORE=";S
 630 PRINT AT 7,0;"PLAY AGAIN?
(YES/NO)"
 640 IF INKEY$="Y" THEN GOTO 70
 650 IF INKEY$<>"N" THEN GOTO 64
0
 660 STOP
1500 IF Z=1 THEN PRINT AT 21,A;"
"
1510 IF Z=2 THEN PRINT AT 21,A;"
"
1520 IF Z=3 THEN PRINT AT 21,A;"
"
1530 RETURN
2000 IF H<1000 AND Z=3 THEN PRIN
T "SORRY, YOU MUST SCORE OVER 10
00 TO PLAY LEVEL 3."
2005 IF H<1000 AND Z=3 THEN GOTO
 110
2010 GOTO 120
5000 CLEAR
5010 SAVE "CATERPILLAR ▒"
5020 RUN
```

AGF

PRICE BREAKTHROUGH

EXCLUSIVE MAIL ORDER SPECIAL OFFER

9.95

AGF Joystick Interface II for Spectrum or ZX81

Now the AGF Interface II is even better value. Since we pioneered the cursor-key interface in October 1982 there are now over 100 games or utility programs with either the AGF option or cursor key controlled – that makes it unbeatable at this new low price.

Still incorporating the all important rear expansion connector which means other peripherals can be connected at the same time i.e Ram Packs, Printers, Speech Units etc, and of course the key replication principle used guarantees this will never conflict electrically with any other add-ons.

When you feel like upgrading to a more flexible joystick system you can claim 12.5% off the purchase price of our Programmable Interface.

All AGF Joystick interfaces will work with any industry standard joysticks using 9-pin 'D' plugs – including QuickShot II and the new Trackball controllers.

Don't settle for less . . . . choose AGF.

Joysticks

The choice is yours. The standard QuickShot is excellent value incorporating suction cups for sure footed precision and a comfortably contoured handle with a convenient top firing button, plus conventional button in the base.

Four feet of cord terminates in the industry standard 9-pin 'D' plug for compatibility with our interfaces or direct connection to a selection of home micros.

The new QuickShot II has improved styling with a trigger type firing control as well as the top firing action and a broader base for greater suction stability. The rapid fire switch, which allows a continuous stream of shells where normally a button press is required for each one, is located in the base for easy access in use.

Cord length and termination is as the standard QuickShot controller.

8.95

QuickShot

QuickShot II

11.95

12.5% discount off our Programmable Interface~valid 1 year on orders over £18

PLUS free m/c arcade game with 1st 100 orders Spectrum only

so HURRY . . . post free both ways

Please allow up to 28 days from receipt of your order although we normally despatch from stock within 7 days. All AGF products are guaranteed for 12 months and are supplied on a 14 day money back undertaking. AGF Hardware, Bognor Regis, West Sussex. Telephone (0243) 823337.

FROM: MR/MRS/MISS

ADDRESS

I ENCLOSE A CHEQUE/POSTAL ORDER PAYABLE TO AGF HARDWARE FOR £

SEND (NO STAMP NEEDED) TO: A.G.F. HARDWARE, DEPT. **PS**,

FREEPOST, BOGNOR REGIS, WEST SUSSEX, PO22 9BR

QTY	ITEM	ITEM PRICE	TOTAL
	AGF JOYSTICK INTERFACE II	9.95	
	QUICKSHOT JOYSTICK	8.95	
	QUICKSHOT II JOYSTICK	11.95	
ONE	DISCOUNT VOUCHER IF ORDER EXCEEDS £18	FREE	
ZX81 ☐	ZX SPECTRUM ☐ Please tick	FINAL TOTAL	

Jeremy Lever

CRAZY BALLOONING

Take part in an aerial assault course.

This is an adaptation of the popular arcade game for the ZX81. You have to guide the balloon around the course of prickly asterisks to reach the area marked 'GOAL' in the top left-hand corner of the maze.

You start at the bottom right-hand corner of the course and the controls you should use to guide your balloon are shown in Fig. 1. Obviously, the idea of the game is to try and get your balloon to the goal without it hitting the asterisks which, in case your bubble!

PROGRAM DESCRIPTION

Below is a breakdown of the program.

You can always take a short cut through one of the two gates in the asterisk wall — but take care, this is not always the easiest route!

Lines 1-85	Initialisation.
Lines 90-185	What key has been pressed?
Lines 193-240	What has the balloon hit?
Lines 255-300	Swing routine.
Lines 400-520	The 'balloon burst' and 'high score' routines.
Lines 550-570	The goal routine
Lines 600-620	The bounce at the start of the game.
Lines 1000-1250	The routine to set up the screen.
Lines 2000-2200	The routine containing the instruction.
Lines 3000-3320	The movement routine.

```
  1 REM *CRAZY BALLOON*
  2 LET S=0
  3 LET HS=50
  4 LET R=0
  5 LET N$="ZX81"
  7 SLOW
 10 CLS
 11 LET K=0
 12 LET X=0
 13 LET BA=3
 15 PRINT "INSTRUCTIONS (Y/N)"
 16 LET U$=INKEY$
 17 IF U$="Y" THEN GOTO 2000
 18 IF U$="N" THEN GOTO 20
 19 GOTO 16
 20 CLS
 21 LET X=0
 25 GOSUB 1000
 27 IF X=16 THEN LET O=0
 30 PRINT AT 0,0;"SCORE ";S
 35 PRINT AT 0,14;N$
 40 PRINT AT 0,25;HS
 45 PRINT AT 20,0;BA;" BALLOONS
LEFT"
 60 LET V=16
 70 LET Y=V+1
 80 LET H=21
 85 LET Z=H
 90 PRINT AT V,H;"O"
100 PRINT AT Y,Z;"I"
110 LET A$=INKEY$
111 FAST
115 PRINT AT 0,6;S
120 IF A$="D" THEN GOTO 3180
130 IF A$="A" THEN GOTO 3220
140 IF A$="W" THEN GOTO 3260
150 IF A$="X" THEN GOTO 3300
160 IF A$="E" THEN GOTO 3000
170 IF A$="Q" THEN GOTO 3050
180 IF A$="C" THEN GOTO 3100
185 IF A$="Z" THEN GOTO 3150
191 LET S=S-1
192 LET S=S+1
193 IF K=1 THEN GOTO 196
194 PRINT AT Y,Z;
195 GOTO 200
196 PRINT AT V,H;
200 LET P=PEEK (256*PEEK 16399+
PEEK 16398)
210 IF P=23 THEN GOTO 400
220 IF P=128 THEN GOTO 550
230 IF P=8 THEN GOSUB 600
231 IF P=131 THEN LET S=S+60
232 IF P=9 THEN LET S=S+40
233 IF K=1 THEN GOTO 239
234 LET K=1
235 GOTO 196
239 LET K=0
240 SLOW
245 PRINT AT V,H;"O"
250 PRINT AT Y,Z;"I"
255 LET X=X+1
260 IF X=5 THEN LET H=H-1
261 IF X=5 THEN PRINT AT V,H+1;
" "
265 IF X=10 THEN LET H=H+1
266 IF X=10 THEN PRINT AT V,H-1
" "
270 IF X=15 THEN LET H=H+1
271 IF X=15 THEN PRINT AT V,H-1
" "
275 IF X=20 THEN LET H=H-1
276 IF X=20 THEN PRINT AT V,H+1
" "
280 IF X=25 THEN LET H=H-1
```

```
281 IF X=25 THEN PRINT AT V,H+1
" "
290 IF X=30 THEN LET H=H+1
291 IF X=30 THEN PRINT AT V,H-1
" "
292 IF X=35 THEN LET X=10
300 GOTO 110
400 PRINT AT V,H;"X"
401 PRINT AT Y,Z;"I"
402 LET BA=BA-1
405 PAUSE 70
410 CLS
420 PRINT AT 10,11;"*B*A*N*G*"
421 SLOW
425 PAUSE 100
426 IF BA<>0 THEN GOTO 20
430 IF S>HS THEN GOTO 460
440 PAUSE 9999
445 LET S=0
447 SLOW
450 GOTO 10
460 CLS
470 PRINT "NAME"
480 INPUT K$
490 IF LEN (K$)>10 THEN GOTO 46
0
495 LET N$=K$
500 LET HS=S
510 LET S=0
520 GOTO 10
550 LET C=INT (RND*1000)
551 SLOW
555 IF C<400 THEN GOTO 550
556 LET S=S+C
557 FOR I=1 TO 10
558 PRINT AT 8,1;"GOAL"
559 PRINT AT 8,1;"GOAL"
560 NEXT I
561 PAUSE 9999
570 GOTO 20
600 LET V=17
610 LET H=21
620 RETURN
1000 PRINT AT 1,0;"**************
*************"
1020 PRINT "**** ***********
*****"
1030 PRINT "***    *******
****"
1040 PRINT "**          **
 *** C"
1050 PRINT "**   *            ***
  ** R"
1060 PRINT "*       **       ******
  ** A"
1070 PRINT "*█████******█*********
 *** Z"
1080 PRINT "*GOAL***          ******
 *** Y"
1090 PRINT "******              ****
  **  "
1100 PRINT "*****       **       *
 *** B"
1110 PRINT "***         ****
 *** A"
1120 PRINT "**        *******
  ** L"
1130 PRINT "**      *         **▒****
***** L"
1140 PRINT "**     *          *   ***
***** O"
1150 PRINT "*      **     **
▒▒▒▒▒▒* O"
1160 PRINT "**          ****     *
   ▒* N"
1170 PRINT "***       ***********
   ▒*"
1180 PRINT "*********************▒
▒▒▒▒▒▒*"
1190 PRINT "**********************
*****"
1210 FOR Q=4 TO 16
1220 PRINT AT Q,26;" "
1230 NEXT Q
1231 IF R=1 THEN GOTO 1250
1232 LET R=R+1
1235 GOTO 1000
1250 RETURN
2000 CLS
2005 PRINT AT 0,9;"CRAZY BALLOON
"
2007 PRINT AT 1,9;"-------------
"
2010 PRINT "MOVE ROUND THE COURS
E AVOIDING"
2015 PRINT
2020 PRINT "THE PRICKLY STARS- *
"
2025 PRINT
2030 PRINT "EXTRA POINTS ARE AWA
RDED FOR"
2035 PRINT
2040 PRINT "GOING THROUGH THE GA
TES- ▄ + ▄"
2045 PRINT
2050 PRINT "THE OBJECT IS TO GET
 TO THE"
2055 PRINT
2060 PRINT "GOAL IN THE TOP LEFT
 HAND"
2065 PRINT
2070 PRINT "CORNER."
2075 PRINT
2077 PRINT "THE CONTROLS ARE:"
2078 PRINT
2080 PRINT AT 18,15;"QWE"
2090 PRINT AT 19,15;"A*D"
2100 PRINT AT 20,15;"ZXC"
2180 PAUSE 9999
2190 CLS
2200 GOTO 20
3000 LET V=V-1
3010 LET H=H+1
3011 LET Y=Y-1
3012 LET Z=Z+1
3015 PRINT AT V+1,H-1;" "
3020 PRINT AT Y+1,Z-1;" "
3025 GOTO 192
3050 LET V=V-1
3060 LET H=H-1
3061 LET Y=Y-1
3062 LET Z=Z-1
3065 PRINT AT V+1,H+1;" "
3070 PRINT AT Y+1,Z+1;" "
3075 GOTO 192
3100 LET V=V+1
3110 LET H=H+1
3111 LET Y=Y+1
3112 LET Z=Z+1
3115 PRINT AT V-1,H-1;" "
3120 PRINT AT Y-1,Z-1;" "
3125 GOTO 192
3150 LET V=V+1
3160 LET H=H-1
3161 LET Y=Y+1
3162 LET Z=Z-1
3165 PRINT AT V-1,H+1;" "
3170 PRINT AT Y-1,Z+1;" "
3175 GOTO 192
3180 LET H=H+1
3185 LET Z=Z+1
3190 PRINT AT V,H-1;" "
3195 PRINT AT Y,Z-1;" "
3200 GOTO 192
3220 LET H=H-1
3225 LET Z=Z-1
3230 PRINT AT V,H+1;" "
3235 PRINT AT Y,Z+1;" "
3240 GOTO 192
3260 LET V=V-1
3265 LET Y=Y-1
3270 PRINT AT V+1,H;" "
3275 PRINT AT Y+1,Z;" "
3280 GOTO 192
3300 LET V=V+1
3305 LET Y=Y+1
3310 PRINT AT V-1,H;" "
3315 PRINT AT Y-1,Z;" "
3320 GOTO 192
```

This is the listing for the program. Crazy Balloon.

TALKING TURKEY

This program has decided that although the English language does not exist, some of its rules do.

Using these rules, the computer tries to invent English words, and manages to do so surprisingly often. Around seven per cent of the output of this program should be real words. The ZX81 uses knowledge of the frequency of occurrence of certain letters in words in English to dictate how often the letters are used in creating randomly generated words.

If you leave this program running for a million years it may well write the Gettysburg Address ('Three score and seven years ago . . .').

PROGRAM LISTING

```
  10 DIM A$(26,23)
  20 LET A$(1)="TNIRSHMGBCDFJKLP
QUAVWXYZ"
  25 LET A$(2)="EAOIU"
  30 LET A$(3)="ETAOISHU"
  35 LET A$(4)=A$(3)
  40 LET A$(5)="ETANRSHMGBCDFJKL
PQUVWXYZ"
  50 LET A$(6)=A$(2)
  60 LET A$(7)="EAOISHU"
  70 LET A$(8)=A$(2)
  80 LET A$(9)="TEONRSHMGBCDFJKL
PQUVWXYZ"
  90 LET A$(10)=A$(2)
 100 LET A$(11)=A$(2)
 110 LET A$(12)=A$(2)
 120 LET A$(13)=A$(2)
 130 LET A$(14)=A$(2)
 140 LET A$(15)="TAONIRSHMGBCDFK
JLPQUVWXYZ"
 150 LET A$(16)=A$(2)
 160 LET A$(17)="U"
 170 LET A$(18)=A$(2)
 180 LET A$(19)="EAOIHU"
 190 LET A$(20)=A$(2)
 200 LET A$(21)="AO"
 210 LET A$(22)=A$(2)
 220 LET A$(23)=A$(2)
 240 LET A$(24)=A$(2)
 250 LET A$(25)=A$(2)
 260 LET A$(26)=A$(2)
 265 LET L=INT (3+RND*INT (RND*4
+1))
 270 LET X=INT (RND*26+1)
 280 FOR I=1 TO L
 290 PRINT CHR$ (X+37),
 300 LET C=CODE A$(X,(INT (RND*I
NT (RND*23)+1)))
 310 IF C=0 THEN GOTO 300
 320 LET X=C-37
 330 NEXT I
 340 IF INKEY$="" THEN GOTO 340
 345 CLS
 350 GOTO 265
```

A Poetry version of the program.

```
   5 REM "POETRY" VERSION
  10 DIM A$(26,23)
  20 LET A$(1)="TNIRSHMGBCDFJKLP
QUAVWXYZ"
  25 LET A$(2)="EAOIU"
  30 LET A$(3)="ETAOISHU"
  35 LET A$(4)=A$(3)
  40 LET A$(5)="ETANRSHMGBCDFJKL
PQUVWXYZ"
  50 LET A$(6)=A$(2)
  60 LET A$(7)="EAOISHU"
  70 LET A$(8)=A$(2)
  80 LET A$(9)="TEONRSHMGBCDFJKL
PQUVWXYZ"
  90 LET A$(10)=A$(2)
 100 LET A$(11)=A$(2)
 110 LET A$(12)=A$(2)
 120 LET A$(13)=A$(2)
 130 LET A$(14)=A$(2)
 140 LET A$(15)="TAONIRSHMGBCDFK
JLPQUVWXYZ"
 150 LET A$(16)=A$(2)
 160■LET A$(17)="U"
 170 LET A$(18)=A$(2)
 180 LET A$(19)="EAOIHU"
 190 LET A$(20)=A$(2)
 200 LET A$(21)="AO"
 210 LET A$(22)=A$(2)
 220 LET A$(23)=A$(2)
 240 LET A$(24)=A$(2)
 250 LET A$(25)=A$(2)
 260 LET A$(25)=A$(2)
 262 FOR G=1 TO 100
 265 LET L=INT (3+RND*INT (RND*4
+1))
 270 LET X=INT (RND*26+1)
 275 SCROLL
 277 FOR H=1 TO RND*5+2
 280 FOR I=1 TO L
 290 PRINT CHR$ (X+37);
 300 LET C=CODE A$(X,(INT (RND*I
NT (RND*23)+1)))
 310 IF C=0 THEN GOTO 300
 320 LET X=C-37
 330■NEXT I
 332 IF L<5 THEN LET L=L+INT (RN
D*2)-INT (RND*2)
 335 PRINT " ";
 336 NEXT H
 337 IF RND>.8 THEN SCROLL
 340 NEXT G
 350 GOTO 262
```

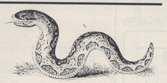
Part of the output.

```
AIBO      YOP     AMA
DTEH'     DEN     SUABA
TOO       LOT     DTAKA
UOTI      VEE     HACA
NEFAIT    NIT     MAT
REE       HUA     INAX
HUAH      RERA    XIO
BIE       TEGO    SOM
YINOR     PEAGH   SITE
YIE       WEF     LUOO
JEC       NUA     SEX
BATU      IEB     PIG
JIT       CET     ZEXE
RIH       NIO     ZOT
OGI       SAT     ZOGAN
RAC       UARIB   BED
IES       TEM     XAN
KITE      SHAG    FENE
LIO       KOR     BAUAN
BAHETA    HEMI    TAMA
KASITA    LUO     DTAC
LU                HAC
```

Gary Nugent

WORMS

If you can worm your way round the machine code, you'll find it's worth it.



This is a two player game, the object being to force your opponent to collide with a wall, your worm or your opponent's worm.

The game uses the full 24 by 32 screen, the walls being the black border around the screen. One worm is made from '+' characters, the other is made from '*' characters. The worms grow as you move around the screen.

Movement is in four directions; up down, left and right. Keys 'W' (up), 'A' (left), 'S' (right) and 'Z' (down) control the '+' worm. Keys 'O' (up), 'K' (left), 'L' (right) and '.' (down) control the '*' worm. A worm is killed if it hits a wall, hits the opponent's worm, collides with itself or moves back on itself. The game also has on-screen scoring for both players. Key 'F' returns control to BASIC and stops the program.

DOWN THE LINE

Looking at the main program. The machine code is held in the line 1 REM statement containing 532 characters. There is an easy way and a hard way to create such a REM statement. The hard way is to type in the 532 characters one after another. The easy way is as follows: create the REM statement in Fig. 1, that is, a REM followed by 100 characters. Edit the line numbers to form lines 2 to 5 and then enter line 6 (a REM followed by two characters) as in Fig. 2. Next, execute the following as direct commands:

POKE 16510,0 (this ensures the line cannot be edited)

```
POKE 16511,22
POKE 16512,2
```

A REM containing 532 characters has now been created, and the machine code program can now be entered using a hex-loader like that in Fig. 3. Incidentally, if PRINT PEEK 16511 + 256*PEEK 16512-2 does not give the answer 532 then you have typed in the REMs wrongly, and you should go through the above procedure again.

After you have entered the machine code program, you can check to see if you have entered it correctly, using the program in Fig. 4. When you are sure you have entered the machine code program correctly, you can now enter the BASIC control program. Save the program first by RUN 200, just in case something goes wrong.

TOO FAST?

To increase the speed of the game, type in the following as direct commands:

POKE 16912,X (Where X is a value between zero and 255. The lower the value, the faster the game.)

```
POKE 16913,0
```

To slow the game down, type in the following:

POKE 16912,X (Where X lies between zero and 255. The greater the value the slower the game.)

POKE 16913,Y ('1' or '2' is usually a big enough value for Y).

If you would like to change the characters which form the worms, then change the '+' by typing in the following:

POKE 16838,C
POKE 16797,C+128 (Inverse of C.)

```
POKE 16870,C
POKE 16880,C
POKE 16905,C
```

where C is the code of the required character.

To change the '*' worm, type in the following:

```
POKE 16850,C
POKE 16875,C
POKE 16900,C
POKE 16910,C
POKE 16813,C+128
```

The program, as published, was originally written for a 16K ZX81, although with the addition of a little bit of machine code it will run on a machine which has 3K of RAM, ie where the display file is not automatically filled with spaces. To make the program run in 3K, the machine code should be placed in a line 2 REM statement.

```
LD HL, 4022              21 22 40
LD (HL),00                  36 00
LD A,00                 3E 00
LD C,18              OE 18
LD B,20 NEXTLN            06 20
RST 16 NEXTCHR          D7
DJNZ NEXTCHR         10 FD
DEC C                   0D
JR NZ,NEXTLN            20 F8
RET                     C9
```

▶

You should also insert this line in the program:

9 RAND USR 17053

The code creates a 24 by 32 display file, which is automatically set up on machines which have more than 3¼K RAM.

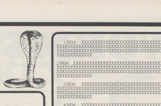


FIG. 1

```
  1 REM ?USR ?
  5 REM ZX81 WORMS
  6 REM FOR 16K ZX81
  7 REM [illegible]
 10 RAND USR 16761
 20 LET A=16514
 30 POKE A,INT (RND*26)+4
 40 POKE A+1,INT (RND*16)+4
 50 POKE A+2,INT (RND*26)+4
 60 POKE A+3,INT (RND*16)+4
 70 POKE A+4,INT (RND*4)+1
 80 POKE A+5,INT (RND*4)+1
 90 IF USR 16824 THEN GOTO 20
100 STOP
200 SAVE "ZX81 WORMS"
210 PRINT AT 9,3;"DO YOU WANT I
NSTRUCTIONS ?"
220 IF INKEY$="N" THEN GOTO 270
230 IF INKEY$<>"Y" THEN GOTO 22
0
235 CLS
240 LIST 500
250 FOR I=1 TO 300
260 NEXT I
270 CLS
280 RUN
500 REM           INSTRUCTIONS
510 REM
520 REM
530 REM    THE OBJECT OF THE
540 REM  GAME IS TO FORCE YOUR
550 REM  OPPONENTS WORM TO
560 REM  COLLIDE WITH ONE OF
570 REM  THE WALLS, OR WITH
580 REM  YOUR OR HIS OWN WORM.
590 REM  KEYS W,A,S,Z CONTROL
600 REM  THE "+".
610 REM  KEYS O,K,L,. CONTROL
620 REM  THE "*".
630 REM  KEY F ENDS THE GAME.
640 REM
```

BASIC CONTROL PROGRAM WHICH CALLS THE MACHINE CODE ROUTINE.

```
1 REM    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXX

2 REM  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXX

3 REM    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXX

4 REM    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXX

5 REM    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXX

6 REM XX
```

Fig. 2.

```
10 LET X= 16514
20 LET AS=""
30 IF AS="" THEN INPUT AS
40 IF AS="S" THEN STOP
50 PRINT AS (TO 2);" ";
60 POKE X, 16*CODE As+CODE AS(2)-476
70 LET X=X+1
80 LET AS=AS(3 TO)
90 GOTO 30
```

Fig. 3.

```
100 FOR I= 16514 TO 17045
110 LET A=PEEK I
120 LET B=INT (A/16)
130 ELT C=A-16*B
140 PRINT TAB 6;   "-----";CHRS(B+28);CHRs(C+28)
150 NEXT I
```

Fig. 4.

```
DATA: 00 00 00 00 00 00 07 03 03
   03 03 03 03 03 03 03 03 84 76
DATA: 80 00 00 00 00 00 00 00 00
   00 05 3F 3D 24 1D 00 3C 34 37
   32 38 85 76
DATA: 80 00 00 00 00 00 00 00 00
   00 82 83 83 83 83 83 83 83 83
   83 83 81 76
DATA: 76
DATA: 76
DATA: 80 00 00 00 00 00 28 34 35
   3E 37 2E 2C 2D 39 00 2C 26 37
   3E 00 33 3A 2C 2A 33 39 76
DATA: 76
DATA: 80 00 00 00 00 00 00 00 00
   00 00 2F 3A 33 2A 00 1D 25 24
   1E FF
LD HL,(D-FILE)   BORDER 2A0C40
LD DE,0021              112100
LD A,80                 3E80
LD BC,2017              011720
INC HL           LOOP1  23
LD (HL),A               77
DJNZ LOOP1              10FC
ADD HL,DE        LOOP2  19
LD (HL),A               77
DEC C                   0D
JR NZ,LOOP2             20FB
LD BC,1F16              01161F
DEC HL           LOOP3  2B
LD (HL),A               77
DJNZ LOOP3              10FC
SBC HL,DE        LOOP4  ED52
LD (HL),A               77
DEC C                   0D
JR NZ,LOOP4             20FA
RET                     C9
LD BC,FFFF       PAUSE  01FFFF
DEC BC           LOOP5  0B
LD A,B                  78
OR C                    B1
JR NZ,LOOP5             20FB
RET                     C9
LD HL,(D-FILE)   CLEAR  2A0C40
LD DE,0023              112300
ADD HL,DE               19
LD C,16                 0E16
LB B,1E          LOOP7  061E
LD (HL),00       LOOP6  3600
INC HL                  23
DJNZ LOOP6              10FB
INC HL                  23
INC HL                  23
INC HL                  23
DEC C                   0D
```

```
JR NZ,LOOP7                 20F3
RET                         C9
LD E,B              RND     58
LD D,00                     1600
PUSH DE                     D5
POP HL                      E1
ADD HL,HL                   29
ADD HL,HL                   29
ADD HL,HL                   29
ADD HL,HL                   29
ADD HL,HL                   29
ADD HL,DE                   19
LD DE,(D-FILE)              ED5B0C40
INC DE                      13
ADD HL,DE                   19
LD B,00                     0600
ADD HL,BC                   09
RET                         C9
LD BC,0021          MOVE    012100
CP 01                       FE01
JR NZ,STEP1                 2002
INC HL                      23
RET                         C9
CP 02               STEP1   FE02
JR NZ,STEP2                 2002
DEC HL                      2B
RET                         C9
CP 03               STEP2   FE03
JR NZ,STEP3                 2002
ADD HL,BC                   09
RET                         C9
SBC HL,BC           STEP3   ED42
RET                         C9
LD A,(HL)           INCSC   7E
CP 9C                       FE9C
JR NZ,STEP4                 2002
LD A,9C                     3E9C
INC A               STEP4   3C
CP A6                       FEA6
JR NZ,STEP5                 2005
LD (HL),9C                  369C
DEC HL                      2B
JR LOOP8                    18EF
LD (HL),A           STEP5   77
RET                         C9
CALL BORDER                 CDF740
LD BC,080A                  010A08
CALL PRINT-AT               CDF508
LD HL,4088                  218840
LD A,(HL)           LOOP9   7E
CP FF                       FEFF
JR Z,STEP6                  2804
RST 16                      D7
INC HL                      23
JR LOOP9                    18F7
LD D,03             STEP6   1603
CALL PAUSE          LOOP10  CD1941
DEC D                       15
JR NZ,LOOP10                20FA
LD BC,0004                  010400
CALL PRINT-AT               CDF508
LD A,95                     3E95
RST 16                      D7
LD A,94                     3E94
RST 16                      D7
LD A,9C                     3E9C
RST 16                      D7
RST 16                      D7
RST 16                      D7
LD C,16                     0E16
CALL PRINT-AT               CDF508
LD A,97                     3E97
RST 16                      D7
LD A,94                     3E94
RST 16                      D7
LD A,9C                     3E9C
RST 16                      D7
RST 16                      D7
RST 16                      D7
RET                         C9
CALL CLEAR                  CD2241
LD BC,(4082)                ED4B8240
CALL RND                    CD3941
LD (4082),HL                228240
LD (HL),15                  3615
LD BC,(4084)                ED4B8440
CALL RND                    CD3941
LD (4084),HL                228440
LD (HL),17                  3617
LD HL,(4082)        NEXTGO  2A8240
LD A,(4086)                 3A8640
```

```
CALL MOVE                   CD4E41
LD (4082),HL                228240
LD A,(HL)                   7E
CP 80                       FE80
JP NC,SCORE1                D28042
CP 15                       FE15
JP Z,SCORE1                 CA8042
CP 17                       FE17
JP Z,SCORE1                 CA8042
LD (HL),15                  3615
LD HL,(4084)                2A8440
LD A,(4087)                 3A8740
CALL MOVE                   CD4E41
LD (4084),HL                228440
LD A,(HL)                   7E
CP 80                       FE80
JP NC,SCORE2                D28B42
CP 17                       FE17
JP Z,SCORE2                 CA8B42
CP 15                       FE15
JP Z,SCORE2                 CA8B42
LD (HL),17                  3617
LD DE,0100                  110001
PUSH DE             SEARCH  D5
CALL KSCAN                  CDBB02
LD B,H                      44
LD C,L                      4D
LD D,C                      51
INC D                       14
LD A,00                     3E00
JR Z.NOCHR                  2855
CALL FINDCHR                CDBD07
LD A,(HL)                   7E
CP 3C                       FE3C
JR NZ,CHAR1                 2004
LD A,04                     3E04
JR DIR1                     1841
CP 34               CHAR1   FE34
JR NZ,CHAR2                 2004
LD A,04                     3E04
JR DIR2                     183E
CP 3F               CHAR2   FE3F
JR NZ,CHAR3                 2004
LD A,03                     3E03
JR DIR1                     1831
CP 1B               CHAR3   FE1B
JR NZ,CHAR4                 2004
LD A,03                     3E03
JR DIR2                     182E
CP 26               CHAR4   FE26
JR NZ,CHAR5                 2004
LD A,02                     3E02
JR DIR1                     1821
CP 30               CHAR5   FE30
JR NZ,CHAR6                 2004
LD A,02                     3E02
JR DIR2                     181E
CP 38               CHAR6   FE38
JR NZ,CHAR7                 2004
LD A,01                     3E01
JR DIR1                     1811
CP 31               CHAR7   FE31
JR NZ,CHAR8                 2004
LD A,01                     3E01
JR DIR                      180E
CP 2B               CHAR8   FE2B
JR NZ,NOCHR                 200D
LD BC,0000                  010000
JR NOCHR                    1808
LD (4086),A         DIR1    328640
JR NOCHR                    1803
LD (4087),A         DIR2    328740
POP DE              NOCHR   D1
DEC DE                      1B
LD A,D                      7A
OR E                        B3
JR NZ,SEARCH                2099
LD A,C                      79
CP 00                       FE00
RET Z                       C8
JP NEXTGO                   C3D341
LD HL,(D-FILE)      SCORE1  2A0C40
LD DE,001B                  111B00
ADD HL,DE                   19
CALL INCSC                  CD5641
RET                         C9
LD H,(D-FILE)       SCORE2  2A0C40
LD DE,0009                  110900
ADD HL,DE                   19
CALL INCSC                  CD6641
RET                         C9
```

P J Kenworthy

BACKGAMMON

Here's a classic game brought up to date with a program for the ZX81 that plays against you and obeys all the rules.

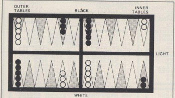


Fig. 1. The starting position for backgammon.

Backgammon is one of the more popular and enduring of board games, probably because it doesn't rely solely on blind chance and the throw of the dice, but requires an appreciation of probabilities and the skill to assess the most effective use of your throws. This program puts a simulation of the backgammon board on your TV screen, and allows you to play against the computer. We aren't pretending that this piece of software could wipe the floor with Omar Sharif but it will put up a creditable fight, and can usually be guaranteed to cause you the maximum difficulty if the dice are with it. The performance is especially good when you realise that there isn't enough memory in a 16K ZX81 for a look-ahead probability tree even one turn deep.

Just to make sure that you don't get tempted into cheating it, the computer checks out everything you enter against the rules of backgammon and rejects invalid moves. The one exception is the rule stating that if a player can only use one of the numbers he has rolled, he must, if possible, use the larger. This was in the copy of rules we obtained, but two backgammon players in the office said they'd never heard of it, so we decided to leave it out. If you play by this rule, it wouldn't require more than about five or six extra lines of code: this is left as an exercise to the reader!

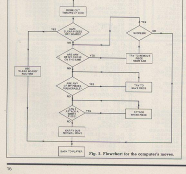

WORK OUT THROWS OF DICE
CAN I CLEAR PIECES OFF BOARD?
YES
NO
USE 'CLEAR BOARD' ROUTINE
ARE ANY OF MY PIECES ON THE BAR?
YES
TRY TO REMOVE PIECE FROM BAR
SUCCEED?
YES
NO
ARE ANY OF MY PIECES VULNERABLE?
YES
TRY TO SAVE PIECE
CAN I ATTACK A WHITE PIECE?
YES
ATTACK WHITE PIECE
CARRY OUT NORMAL MOVE
BACK TO PLAYER

Fig. 2. Flowchart for the computer's moves.

SIMULATION

The working of the program is based on the two position arrays W( ) and B( ), which correspond to the points on the backgammon board. The board consists of 24 positions, or points, each of which may hold any number of pieces (of the same colour) up to the total number in play. The array elements B(1)-B(24) hold the number of black pieces per position, while the corresponding information for the white player is held in W(1)-W(24). For example, at the start of the game B(1) is set to two, since there are two black pieces at position A on the board. During the course of the game these array elements are constantly changing.

Two further elements of each array are used in the game, these being W(31), B(31), W(35) and B(35). The first two hold the number of pieces each player has on the bar, while the second two hold the number of pieces each player has taken off the board at the end of the game.

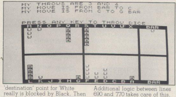
The computer uses the position arrays to work out its move. For every move that the computer makes, the routine shown in the flowchart of figure 2 will be carried out. To give an example, suppose one of the computer's throws is a five. The program will look at each black piece, take the index of its position in the array and add five to it. It can then see the effect of moving the piece five places — whether the move is legal, whether a white piece could be taken, whether a position could be fortified and so on. If the computer is satisfied that the move is a good one, it is carried out.

Things get more complicated when the computer has to check whether a move that you have made is legal. The player's dice throws are stored in a third array, T( ), to simplify the checking routines by using FOR-NEXT loops. If the dice throws are not equal, the two numbers are put into T(1) and T(2) in descending order (using T(5) as a temporary store) and the remaining array elements made zero. Otherwise doubles have been thrown, and the first four elements of the array are all made equal, T(5) again ending up as zero.

If a player's move is judged to be valid, the dice throw corresponding to that move must be tagged as 'used' by setting it to an 'impossible' value. We decided that the answer was to use 42!

Since the three arrays completely describe the state of the game with respect to piece positions and dice in play, the computer can run through a series of tests involving AND, OR and the arithmetic operators, one test per rule. As an example, the test that a player is genuinely unable to move is made between lines 1370 and 1460. For each unused throw of the dice and each point on the board, line 1400 checks that the corresponding 'destination' point for White really is blocked by Black. Then line 1410 checks that, if White is on the bar, he genuinely cannot move a piece back into play.

An added complication is introduced when White starts to bear off, since it is possible to use a dice throw greater than the distance the piece is moved. Additional logic between lines 690 and 770 takes care of this.



```
MY THROWS ARE 3 AND 4
MY MOVE IS FROM BAR TO C
MY MOVE IS FROM C TO G BAR

PRESS ANY KEY TO THROW DICE
```

PLAY THE GAME

It is assumed that the user of this program already knows how to play backgammon — it is not a teaching aid. The following instructions are merely to show

VARIABLE	FUNCTION
C (= M)	Random number used for computer's first throw
D (= V)	Random number used for computer's second throw
G$,GG	Player's move (from)
H$,HH	Player's move (to)
DD	Distance of a player's move
F,J,K,L, N,S,Q,T, U,X,CC,TT	General purpose loop counters and flags
W( )	Position array for white pieces
B( )	Position array for black pieces
W(31)	White pieces out of play on the bar
B(31)	Black pieces out of play on the bar
W(35)	White pieces moved off the board
B(35)	Black pieces moved off the board
T( )	Array containing White's dice throws

PROGRAM STRUCTURE

10-190	Dimension and initialise arrays. Initialise variables.
200	Generate a random number and decide who goes first.
220-350	Throw random numbers for player's dice and display them. Sort them into descending order for testing later. Check for doubles and amend loop counter and array if necessary.

Inputting player's moves

370-540	Display appropriate prompts and accept input. Convert strings to character codes, then subtract offset to put them in the range 1-24. Make preliminary validity checks, and alter GG or HH to appropriate values if "BAR" or "REST" have been input.
550-610	If White is bearing off, check that move is

how to play the game with the ZX81.

The player always plays White and should move in an anti-clockwise direction from X to A (see the photographs). When the program is first run, the computer may move first: if this has happened, the computer's dice throws and moves are displayed at the top of the screen. To throw you own dice, press any key, and after eight seconds or so the computer will display the board and your throws. This will happen anyway if the computer has allowed you to move first. (If you would prefer the computer to automatically throw your dice for you after an eight-second pause, delete lines 3190-3210 and change line 3320 to

```
3220 PAUSE 400
```

Pressing any key will terminate the pause early, and you can change the delay to suit your needs).

The player will then be asked for his move with the prompt "YOUR MOVE FROM" — you should enter the letter of the point you want to move from, then press NEWLINE. The prompt "TO" then appears, and the destination point is entered in a similar way. If the move is invalid, the computer will say so and repeat the above procedure.

If you can't move you should reply "NO" to the prompt "YOUR MOVE FROM" and the computer will make its next set of moves. If you've made a mistake in entering the "MOVE FROM" letter, you can cancel the move by typing "NO" in response to the "TO" prompt. The computer will then go back and ask for the move again. Once the computer has accepted a move as valid, you cannot change it.

When moving one of your pieces back into play from the bar, enter "BAR" in response to the first prompt. If you are moving one of your pieces off the board at the end of the game, enter "REST" in response to the prompt "TO".

If you throw a double, you are required to make four sets of moves.

	valid because no pieces are on the bar or off the inner table.
620-630	Set move distance variable.
620-680	Test that move distance corresponds to one of the dice throws. If White isn't bearing off, jump directly to 840 to test flag.
690-830	Check that, if player is bearing off with a dice throw greater than required, no White pieces are on higher point. Check that a dice throw exists to permit the required move, and set DD to this value ready for the next routine.
840-910	Check that a dice throw exists to permit the required move, and change the array values to register White's move if it's valid.
920-950	Check for a win by White and print appropriate messages. Stop program.
960-980	Check if any of Black's pieces have been taken and alter array if necessary. Loop for next move.
990	Go to Black's move.
1000-1020	Routine to display invalid move.

Computer's move

1030-1070	Subroutine to check whether the computer can start moving its pieces off the board (similar to routine for White above).
1080-1130, 1250-1300	Computer decides whether to make an ordinary move or move pieces off the board.
1140-1160	Remove piece from board with first move.
1310-1350	Remove piece from board with second move.
1170-1240	Display the move and change the array. Check if Black has won and display appropriate messages.
1360-1460	Entry point to computer's move routine if White has been unable to move. This is checked and a jump made to the invalid message routine if necessary. Otherwise fall through.
1470-1580	Computer's throws calculated and displayed. Loop counter altered if doubles thrown.
1590	Check to see whether any of computer's pieces are on the bar. If so, go to the bar-clearing routine at line 3120.
1600-1620	Call subroutine to check if computer can bear off. Go to bearing-off routine at 1080 if so.
1630	Decide whether to play defensively or not (go to line 1670 if not).
1640-1660	Check if any pieces are vulnerable. Go to routine at 2010 if any are.
1670-1730	First move (attack white pieces). Check to see if a black piece can move from the point in question. If a white pieces is available for taking on that spot, go to the 'take piece' routine at 2870. An identical routine for the second move is at lines 1870-1930.
1750-1800	First move (if no white pieces can be attacked). Check if a black piece can be moved legally and do so. An identical routine for the second move is at lines 1940-2000.
2010-2150	Routines for the saving of exposed pieces. Check to see if a white piece can be taken with either the first move (2020), or the second move (2050), or with both moves added together (2070). If a white piece can be taken in any of these ways, the appropriate action is taken. If no white piece is exposed,

a check is carried out to see if an ordinary move is possible for the piece (2100 and 2130). If one is found it is implemented.

Drawing the board

2160-2240	Plot in frame.
2250-2260	Print top and bottom of frame using inverse video characters.
2270-2600	Display pieces on board using PRINT AT. If more than five pieces are on a point, replace top piece with number of pieces.
2610-2740	If either player has pieces on the bar, display them at the side of the board.

Registering Black's Moves

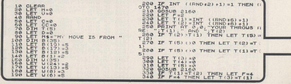
2750-2860	Registering of move if no white piece has been taken (2750-2800 for the first move, 2810-2860 for the second). Moves displayed on screen by lines 2780 and 2840.
2870-3020	Registering of move if white piece has been taken (2870-2940 for the first move, 2950-3020 for the second). Moves displayed on screen by lines 2920 and 3000. These moves differ from the previous set because W(31) is incremented by one and the array position that the white piece occupied is decremented by one. The player is informed that a piece has been taken by the message "BAR".
3030-3110	This routine is used if the computer has used both its throws together to take a white piece.
3120-3150	If the computer has a piece or pieces on the bar, this routine decides the printout format to be used. Line 3250 is chosen for the first move, line 3370 for the second. If the computer is unable to move, line 3150 ensures that the appropriate message is printed once only.
3160	If the computer has thrown a double, this line sends the computer back to the beginning of the movement routine to make its second two moves.
3170	Displays the board with the new moves on it.
3190-3240	Waits for the player to press a key before continuing with the next throw of the dice. May be altered to a PAUSE statement for automatic play.
3250-3480	Register moves if the computer has brought a piece back into play from the bar (3250-3360 for the first move, 3370-3480 for the second). If a white piece has been taken too, the routines at 3330 and 3450 are called.

Should either you or the computer get more than five pieces on any point, a complication arises because of the restricted height of the ZX81 display. We got round this by replacing the top piece on the point with the number of pieces on the point (the photographs make this clear).

CONVERSIONS

Conversion to other, more sophisticated machines should prove no problem at all. The initial board position could be set up by using READ-DATA statements. CLEAR (line 10) may need a figure after it on some machines. If you use the PAUSE version, the delay could be obtained with a FOR-NEXT loop. RAND need not be used on some machines, but may be found in the form RANDOMIZE on others. CODE is normally found in the form ASC( ), and a further complication is introduced by the non-standard ZX81 character set. On a machine using standard ASCII codes, the offset 37 in lines 390, 460, 1190, 2780, 2840, 2920, 3000, 3080, 3270, and 3390 would have to be changed to 64. RND is usually RND(0) or RND(1).

As far as the screen display goes, much better simulations of the board should be possible on machines like the Dragon 32, BBC and ZX Spectrum due to the higher resolution and colour. If you have a machine which doesn't support PRINT AT or a similar statement, you could adapt the system we used in The Valley with [HOME] and cursor movement strings.



```
  10 CLEAR
  20 LET M=0
  30 LET V=0
  40 RAND
  50 LET C=0
  60 LET CC=0
  70 DIM T(5)
  80 LET D=0
  90 LET M$="MY MOVE IS FROM "
 100 DIM B(35)
 110 LET B(19)=5
 120 LET B(17)=3
 130 LET B(12)=5
 140 LET B(1)=2
 150 DIM W(35)
 160 LET W(24)=2
 170 LET W(13)=5
 180 LET W(8)=3
 190 LET W(6)=5
```

```
 200 IF INT ((RND*2)+1)=1 THEN G
OTO 1470
 210 GOSUB 2160
 220 LET F=2
 230 LET T(1)=INT ((RND*6)+1)
 240 LET T(2)=INT ((RND*6)+1)
 250 PRINT AT 0,0;"YOUR THROWS A
RE: ";T(1);" AND ";T(2)
 260 IF T(2)>T(1) THEN LET T(5)=
T(2)
 270 IF T(5)<>0 THEN LET T(2)=T(
1)
 280 IF T(5)<>0 THEN LET T(1)=T(
5)
 290 LET T(3)=0
 300 LET T(4)=0
 310 LET T(5)=0
 320 GOSUB 2160
 330 IF T(1)=T(2) THEN LET F=4
 340 IF F=4 THEN LET T(3)=T(1)
 350 IF F=4 THEN LET T(4)=T(1)
 360 FOR L=1 TO F
 370 PRINT AT L,0;"YOUR MOVE FRO
M ";
 380 INPUT G$
 390 LET GG=CODE G$-37
 400 IF G$="NO" THEN GOTO 1360
 410 IF G$="BAR" THEN LET GG=31
 420 PRINT G$;" TO ";
 430 INPUT H$
 440 IF H$="NO" THEN PRINT AT L,
1;"                            "
 450 IF H$="NO" THEN GOTO 370
 460 LET HH=CODE H$-37
 470 PRINT H$
 480 IF G$="REST" THEN GOTO 1000
 490 IF H$="BAR" THEN GOTO 1000
 500 IF G$=H$ THEN GOTO 1000
 510 IF W(31)>0 AND GG<>31 THEN
GOTO 1000
 520 IF B(HH)>1 AND H$<>"REST" T
HEN GOTO 1000
 530 IF W(GG)-1<0 THEN GOTO 1000
 540 IF H$="REST" THEN LET HH=35
 550 LET Q=0
 560 IF HH<>35 THEN GOTO 620
 570 FOR S=7 TO 24
 580 IF W(S)>0 OR W(31)>0 THEN L
ET Q=1
 590 IF Q=1 THEN LET S=25
 600 NEXT S
 610 IF Q=1 THEN GOTO 1000
 620 LET DD=GG-HH
 630 IF GG=31 THEN LET DD=25-HH
 640 LET Q=1
 650 FOR T=1 TO 4
 660 IF DD=T(T) THEN LET Q=0
 670 NEXT T
 680 IF HH<>35 THEN GOTO 840
 690 LET Q=0
 700 FOR T=1 TO 4
 710 IF T(T)=42 OR T(T)=0 THEN G
OTO 760
 720 FOR S=GG+1 TO 6
 730 IF GG<T(T) AND W(S)>0 THEN
LET Q=1
 740 IF GG=T(T) THEN LET Q=0
 750 NEXT S
 760 NEXT T
 770 IF Q=1 THEN GOTO 1000
 780 LET DD=0
 790 FOR T=1 TO 4
 800 IF T(T)=42 OR T(T)=0 THEN G
OTO 820
 810 IF T(T)>=GG THEN LET DD=T(T
)
 820 NEXT T
 830 IF DD=0 THEN GOTO 1000
 840 IF Q=1 THEN GOTO 1000
 850 FOR T=1 TO 4
 860 IF T(T)=42 OR T(T)=0 THEN G
OTO 890
 870 IF DD=T(T) THEN LET T(T)=42
 880 IF T(T)=42 THEN LET T=6
 890 NEXT T
 900 LET W(GG)=W(GG)-1
 910 LET W(HH)=W(HH)+1
 920 IF W(35)=15 THEN PRINT "YOU
WIN ";
 930 IF W(35)=15 AND B(35)=0 AND
 B(31)=0 THEN PRINT "BY A GAMMON
"
 940 IF W(35)=15 AND B(35)=0 AND
 B(31)>0 THEN PRINT "BY A BACKGA
MMON"
 950 IF W(35)=15 THEN STOP
 960 IF B(HH)=1 AND HH<>35 THEN
LET B(31)=B(31)+1
 970 IF B(HH)=1 THEN LET B(HH)=B
(HH)-1
 980 NEXT L
 990 GOTO 1470
1000 PRINT "INVALID MOVE"
1010 PRINT AT L,1;"
               "
1020 GOTO 370
1030 FOR S=1 TO 18
1040 IF B(S)>0 OR B(31)>0 THEN L
ET Q=1
1050 IF Q=1 THEN RETURN
1060 NEXT S
1070 RETURN
1080 IF C=0 THEN GOTO 1250
1090 FOR X=19 TO 24
1100 LET N=X
1110 IF X+C<25 AND B(X)>0 THEN G
OSUB 2750
1120 IF C=0 THEN GOTO 1250
1130 NEXT X
1140 FOR X=(24-(C-1)) TO 24
1150 IF B(X)>0 THEN GOTO 1170
1160 NEXT X
1170 LET B(X)=B(X)-1
1180 LET B(35)=B(35)+1
1190 PRINT "I HAVE MOVED ";CHR$
(X+37);" OFF THE BOARD"
1200 IF B(35)=15 THEN PRINT "I W
IN ";
1210 IF B(35)=15 AND W(35)=0 AND
 W(31)=0 THEN PRINT "BY A GAMMON
"
1220 IF B(35)=15 AND W(35)=0 AND
 W(31)>0 THEN PRINT "BY A BACKGA
MMON"
1230 IF B(35)=15 THEN GOSUB 2160
1240 IF B(35)=15 THEN STOP
1250 IF D=0 THEN GOTO 3160
1260 FOR X=19 TO 24
1270 LET N=X
1280 IF X+D<25 AND B(X)>0 THEN G
OSUB 2810
1290 IF D=0 THEN GOTO 3160
1300 NEXT X
1310 FOR X=(24-(D-1)) TO 24
1320 IF B(X)>0 THEN GOTO 1340
1330 NEXT X
1340 LET D=0
1350 GOTO 1170
1360 PRINT
1370 FOR S=1 TO 24
1380 FOR T=1 TO 4
1390 IF T(T)=42 OR T(T)=0 THEN G
OTO 1430
1400 IF B(S)<2 AND W(S+T(T))>0 A
ND W(31)=0 THEN LET Q=1
1410 IF W(31)>0 AND B(25-T(T))<2
 THEN LET Q=1
1420 IF Q=1 THEN LET T=5
1430 NEXT T
1440 IF Q=1 THEN LET S=25
1450 NEXT S
1460 IF Q=1 THEN GOTO 1000
1470 LET C=INT ((RND*6)+1)
1480 LET X=1
1490 LET M=C
1500 CLS
1510 LET TT=1
1520 LET D=INT ((RND*6)+1)
1530 LET V=D
1540 IF M=V THEN LET X=2
1550 PRINT "MY THROWS ARE ";C;"
AND ";D
1560 FOR U=1 TO X
1570 LET C=M
1580 LET D=V
1590 IF B(31)>0 THEN GOTO 3120
1600 LET Q=0
1610 GOSUB 1030
1620 IF Q=0 THEN GOTO 1080
1630 IF INT ((RND*2)+1)=1 THEN G
OTO 1670
1640 FOR N=1 TO 24
1650 IF B(N)=1 THEN GOTO 2010
1660 NEXT N
1670 IF C=0 THEN GOTO 1810
1680 FOR N=1 TO 24
1690 IF B(N)=0 THEN NEXT N
1700 IF N+C>24 THEN NEXT N
1710 IF W(N)>1 THEN NEXT N
1720 IF W(N+C)=1 THEN GOSUB 2870
1730 IF C=0 THEN GOTO 1810
1740 NEXT N
1750 FOR N=1 TO 24
1760 IF B(N)=0 THEN NEXT N
1770 IF W(N)>1 THEN NEXT N
1780 IF W(N+C)<2 THEN GOSUB 2750
1790 IF C=0 THEN GOTO 1810
1800 NEXT N
1810 IF D=0 THEN GOTO 3140
1820 GOSUB 1030
1830 IF Q=0 THEN GOTO 1080
1840 LET TT=TT+1
1850 IF TT>1 THEN GOTO 1870
1860 GOTO 1640
1870 FOR N=1 TO 24
1880 IF B(N)=0 THEN NEXT N
1890 IF N+D>24 THEN GOTO 1940
1900 IF W(N)>1 THEN NEXT N
```

```
1910 IF W(N+D)=1 THEN GOSUB 2950
1920 IF D=0 THEN GOTO 3140
1930 NEXT N
1940 FOR N=1 TO 24
1950 IF B(N)=0 THEN NEXT N
1960 IF W(N)>1 THEN NEXT N
1970 IF N+D>24 THEN STOP
1980 IF W(N+D)<2 THEN GOSUB 2810
1990 IF D=0 THEN GOTO 3140
2000 NEXT N
2010 IF C=0 THEN GOTO 2040
2020 IF W(N+C)=1 THEN GOSUB 2870
2030 IF C=0 THEN GOTO 1660
2040 IF D=0 THEN GOTO 2060
2050 IF W(N+D)=1 THEN GOSUB 2950
2060 IF D=0 THEN GOTO 1660
2070 IF W(N+C+D)=1 THEN GOSUB 30
30
2080 IF C=0 AND D=0 THEN GOTO 31
40
2090 IF C=0 THEN GOTO 2120
2100 IF W(N+C)=0 THEN GOSUB 2750
2110 IF C=0 THEN GOTO 1660
2120 IF D=0 THEN GOTO 2140
2130 IF W(N+D)=0 THEN GOSUB 2810
2140 IF D=0 AND C=0 THEN GOTO 31
40
2150 GOTO 1660
2160 FOR N=1 TO 62
2170 PLOT N,15
2180 NEXT N
2190 FOR N=1 TO 30
2200 PLOT 0,N
2210 PLOT 63,N
2220 PLOT 50,N
2230 PLOT 25,N
2240 NEXT N
2250 PRINT AT 21,0;" L K J I H G
 F E D C B A   BAR "
2260 PRINT AT 6,0;" M N O P Q R
 S T U V W X   BAR "
2270 FOR K=1 TO 12
2280 IF W(K)=0 AND B(K)=0 THEN G
OTO 2430
2290 IF W(K)>0 THEN GOTO 2370
2300 FOR J=1 TO B(K)
2310 LET S=J
2320 IF S>5 THEN LET S=5
2330 PRINT AT 21-S,25-(K*2);"█"
2340 IF J>5 THEN PRINT AT 21-S,2
5-(K*2);J
2350 NEXT J
2360 GOTO 2430
2370 FOR J=1 TO W(K)
2380 LET S=J
2390 IF S>5 THEN LET S=5
2400 PRINT AT 21-S,25-(K*2);"W"
2410 IF J>5 THEN PRINT AT 21-S,2
5-(K*2);J
2420 NEXT J
2430 NEXT K
2440 FOR K=13 TO 24
2450 IF W(K)=0 AND B(K)=0 THEN G
OTO 2600
2460 IF W(K)>0 THEN GOTO 2540
2470 FOR J=1 TO B(K)
2480 LET S=J
2490 IF S>5 THEN LET S=5
2500 PRINT AT S+6,((K-13)*2)+1;"
█"
2510 IF J>5 THEN PRINT AT S+6,((
K-13)*2)+1;J
2520 NEXT J
2530 GOTO 2600
2540 FOR J=1 TO W(K)
2550 LET S=J
2560 IF S>5 THEN LET S=5
2570 PRINT AT S+6,((K-13)*2)+1;"
W"
2580 IF J>5 THEN PRINT AT S+6,((
K-13)*2)+1;J
2590 NEXT J
2600 NEXT K
2610 IF W(31)=0 AND B(31)=0 THEN
 RETURN
2620 IF B(31)=0 THEN GOTO 2690
2630 FOR J=1 TO B(31)
2640 IF J<6 THEN PRINT AT 21-J,2
8;"█"
2650 IF J>5 AND J<11 THEN PRINT
AT 26-J,29;"█"
2660 IF J>10 THEN PRINT AT 31-J,
27;"█"
2670 NEXT J
```

```
2680 IF W(31)=0 THEN RETURN
2690 FOR J=1 TO W(31)
2700 IF J<6 THEN PRINT AT J+6,28
;"W"
2710 IF J>5 AND J<11 THEN PRINT
AT J+1,29;"W"
2720 IF J>10 THEN PRINT AT J-4,2
7;"W"
2730 NEXT J
2740 RETURN
2750 IF N+C>24 THEN RETURN
2760 LET B(N)=B(N)-1
2770 LET B(N+C)=B(N+C)+1
2780 PRINT M$;CHR$ (N+37);" TO "
;CHR$ (N+C+37)
2790 LET C=0
2800 RETURN
2810 IF N+D>24 THEN RETURN
2820 LET B(N)=B(N)-1
2830 LET B(N+D)=B(N+D)+1
2840 PRINT M$,CHR$ (N+37);" TO "
;CHR$ (N+D+37)
2850 LET D=0
2860 RETURN
2870 IF N+C>24 THEN RETURN
2880 LET B(N)=B(N)-1
2890 LET B(N+C)=B(N+C)+1
2900 LET W(31)=W(31)+1
2910 LET W(N+C)=W(N+C)-1
2920 PRINT M$,CHR$ (N+37);" TO "
;CHR$ (N+C+37);" BAR"
2930 LET C=0
2940 RETURN
2950 IF N+D>24 THEN RETURN
2960 LET B(N)=B(N)-1
2970 LET B(N+D)=B(N+D)+1
2980 LET W(31)=W(31)+1
2990 LET W(N+D)=W(N+D)-1
3000 PRINT M$;CHR$ (N+37);" TO "
;CHR$ (N+D+37);" BAR"
3010 LET D=0
3020 RETURN
3030 IF N+C+D>24 THEN RETURN
3040 LET B(N)=B(N)-1
3050 LET B(N+C+D)=B(N+C+D)+1
3060 LET W(31)=W(31)+1
3070 LET W(N+C+D)=W(N+C+D)-1
3080 PRINT M$;CHR$ (N+37);" TO "
;CHR$ (N+C+D+37)
3090 LET C=0
3100 LET D=0
3110 RETURN
3120 IF W(C)<2 THEN GOTO 3250
3130 IF W(D)<2 THEN GOTO 3370
3140 IF C>0 AND D>0 AND B(31)>0
THEN LET CC=CC+1
3150 IF CC=1 THEN PRINT "I HAVE
BEEN UNABLE TO MOVE"
3160 NEXT U
3170 GOSUB 2160
3180 LET CC=0
3190 PRINT AT 5,0;"PRESS ANY KEY
 TO THROW DICE"
3200 SLOW
3210 IF INKEY$="" THEN GOTO 3210
3220 FAST
3230 CLS
3240 GOTO 220
3250 LET B(31)=B(31)-1
3260 LET B(C)=B(C)+1
3270 PRINT M$;"BAR TO ";CHR$ (C+
37);
3280 IF W(C)=0 THEN PRINT
3290 IF W(C)=1 THEN GOSUB 3330
3300 LET C=0
3310 IF B(31)>0 THEN GOTO 3130
3320 GOTO 1650
3330 LET W(C)=W(C)-1
3340 LET W(31)=W(31)+1
3350 PRINT " BAR"
3360 RETURN
3370 LET B(31)=B(31)-1
3380 LET B(D)=B(D)+1
3390 PRINT M$;"BAR TO ";CHR$ (D+
37);
3400 IF W(D)=0 THEN PRINT
3410 IF W(D)=1 THEN GOSUB 3450
3420 LET D=0
3430 IF B(31)>0 THEN GOTO 3140
3440 GOTO 1640
3450 LET W(D)=W(D)-1
3460 LET W(31)=W(31)+1
3470 PRINT " BAR"
3480 RETURN
```

D J Munroe

3 D MOVER

Creating a three-dimensional image on your 16K ZX81 is not as difficult as you might imagine.

I have been interested for some time in trying to get a program to RUN on the ZX81 which displayed a solid object rotating in space giving a three-dimensional effect.

The program, once typed in, requires a full five minutes RUNing time to complete the POKEing of each of the six picture frames. The program will then automatically begin. The display is quite realistic — there is no screen flicker or jumping — and shows a rotating rectangle, spinning on a fixed axis. The rotational effect lasts for around 30 seconds and looks something like the illustration shown in figure 1.

SHAPES OF THINGS

The shape of the figure is easily altered by changing the co-ordinates, X( ) and Y( ) in lines 140 to 600. (You'll need to do this on plotting paper though.)

One restriction of the program is that only the top half (lines 0 to 10) can be used. When I first developed the program, I tried to use the whole screen but I ran out of memory (even with the 16K RAM Pack attached!).

Once you have the program entered into the machine, it should be SAVEd before you attempt to RUN it. Once it has been RUN, any new instructions have to be typed in 'blind' as the PRINT position has been moved down below line 23 of the screen. If you want to re-RUN the program, you should type in 'GOTO 1000'; this will also have to be done 'blind'.

As a final point, the insertion of a '£' sign at lines 630 and 650 is to end each frame with a code signal for the operating machine code to recognise that end of frame, and the end of the last (the sixth) frame.

```
1E FF      LD E,255           Initialisation to repeat
                              cycle 255 times; can be
                              shortened by POKEing
                              smaller numbers to
                              address 30001.
1D         DEC E
7B         LD A,E
FE 00      CP 0
C8         RET IF ZERO
2A 0C 40   LD HL(16396)       Start of display.
01 5A 75   LD BC 30042        Set address of first data
                              bit of first frame.
0A         LD A(BC)
77         LD(HL)A            Prints it.
23         INC HL             Next screen position.
03         INC BC             Next data bit.
0A         LD A(BC)
FE 0C      CP 12              Compare it to '£'.
20 F8      JR NZ 8 steps back If not '£', print it and
                              continue with next data.
03         INC BC             Next data bit.
0A         LD A(BC)
FE 0C      CP 12              Compare it to '£'.
28 E6      JR Z 26 steps back If it is '£', return to start
                              and repeat the cycle. If
                              not '£', delay after
                              printing first frame.
16 02      LD D,02            Delay — POKEing larger
15         DEC D              number than two at
7A         LD A,D             address 30029 will reduce
FE 00      CP 0               the speed of rotation.
20 FA      JR NZ 6 steps back
2A 0C 40   LD HL(16396)       GO TO screen and start
0A         LD A(BC)           again.
18 E4      JR 28 steps back
```

The machine code part of the listing.

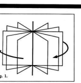


Fig. 1.

BASIC NOTES

Here is a breakdown of the BASIC part of the listing:

Lines 140-600	Initialisation — co-ordinates of the corners of each wire frame (there are six frames altogether).
SUB 4010	Enters the operating machine code starting at address 30000 from A$. Leaves A$ as an empty string.
SUB 1500	SUB 2500 and SUB 3000. Prints the picture of each of the six frames in turn.
Lines 660-800	Convert the picture frame into Hex code and store code in a character string (A$).
SUB 4070	POKEs the Hex code into memory starting at the address after the operating code (address 30042 and leaves A$ as an empty string ready for the next frame.
Lines 1010-1060	Calls up the display with RAND USR 30000.

Listing for the program. 3-D Mover.

```
  20 CLS
  30 REM "3DMOVER"
  40 FAST
  50 DIM X(24)
  60 DIM Y(24)
  70 LET P=0
  80 LET B$=""
  90 LET C$=""
 100 LET D$=""
 110 LET E$=""
 120 LET F$=""
 130 LET G$=""
 140 LET X(1)=20
 150 LET X(2)=40
 160 LET X(3)=20
 170 LET X(4)=40
 180 LET X(5)=23
 190 LET X(6)=37
 200 LET X(7)=23
 210 LET X(8)=37
 220 LET X(9)=26
 230 LET X(10)=34
 240 LET X(11)=26
 250 LET X(12)=34
 260 LET X(13)=31
 270 LET X(14)=30
 280 LET X(15)=31
 290 LET X(16)=30
 300 LET X(17)=26
 310 LET X(18)=34
 320 LET X(19)=26
 330 LET X(20)=34
 340 LET X(21)=23
 345 LET X(22)=37
 350 LET X(23)=23
 360 LET X(24)=37
 370 LET Y(1)=39
 380 LET Y(2)=39
 390 LET Y(3)=26
 400 LET Y(4)=26
 410 LET Y(5)=42
 420 LET Y(6)=36
 430 LET Y(7)=29
 440 LET Y(8)=23
 450 LET Y(9)=43
 460 LET Y(10)=35
 470 LET Y(11)=30
 480 LET Y(12)=22
 490 LET Y(13)=43
 500 LET Y(14)=35
 510 LET Y(15)=30
 520 LET Y(16)=22
 530 LET Y(17)=35
 540 LET Y(18)=43
 550 LET Y(19)=22
 560 LET Y(20)=30
 570 LET Y(21)=36
 580 LET Y(22)=42
 590 LET Y(23)=23
 600 LET Y(24)=29
 605 GOSUB 4010
 610 FOR J=1 TO 21 STEP 4
 620 GOSUB 1500
 630 PRINT AT 10,31;"£"
 640 LET P=P+1
 650 IF P=6 THEN PRINT AT 10,30;
"£"
 660 LET A=PEEK 16396+256*PEEK 1
6397
 665 LET A$=""
 670 FOR B=0 TO 363
 680 LET S=PEEK (A+B)
 690 LET H=INT (S/16)
 700 LET L=(S/16-H)*16
 710 LET L$=CHR$ (L+28)
 720 LET H$=CHR$ (H+28)
 730 LET A$=A$+H$+L$
 790 NEXT B
 800 CLS
 810 GOSUB 4070
 820 NEXT J
1010 CLS
1020 SLOW
1040 RAND USR 30000
1060 STOP
1500 FOR I=J TO J+3 STEP 2
1510 GOSUB 2500
1520 NEXT I
1530 FOR I=J TO J+1
1540 GOSUB 3000
1550 NEXT I
1560 RETURN
2500 LET M=(Y(I)-(Y(I+1)))/(X(I)
-(X(I+1)))
2510 LET C=Y(I)-(M*X(I))
2520 FOR X=X(I) TO X(I+1)
2530 PLOT X,(M*X)+C
2540 NEXT X
2550 RETURN
3000 FOR Y=Y(I+2) TO Y(I)
3010 PLOT X(I),Y
3020 NEXT Y
3024 FOR Y=22 TO 43
3026 PLOT 30,Y
3028 NEXT Y
3030 RETURN
4010 REM "STARTING ADDRESS=30000
"
4020 LET A$="7530"
4030 GOSUB 5000
4060 LET A$="1EFF1D7BFE00C82A0C4
0015A750A7723030AFE0C20F0030AFE0
C28E61602157AFE0020FA2A0C400A18E
4"
4070 IF A$="" THEN RETURN
4100 POKE Z,16*CODE A$+CODE A$(2
)-476
4110 LET Z=Z+1
4120 LET A$=A$(3 TO )
4130 GOTO 4070
5000 LET Z=4096*CODE A$+256*CODE
 A$(2)+16*CODE A$(3)+CODE A$(4)-
122332
5010 RETURN
5020 STOP
```

Adam Waring

JUST ANOTHER BRICK

Did anyone ever say to you "Gee you're a real brick"? Did it make you see red? Well now give vent to your anger . . . Demolish walls and draw lines round your ZX81.

The wall is closing in on you. There's no escape. Can you destroy it before it destroys you? The answer, quite simply, is NO. The wall is endless. Your 'V' is moved rapidly from left to right at the top of the screen. The wall scrolls up towards you. Your only chance is to hurl yourself at the wall in the hope of dislodging some bricks.

The game is fast for a BASIC program. This is because characters are POKEd into the display file — much faster than PRINT AT, especially when dealing with numbers. Your score is POKEd on the bottom line of the screen, making use of the full 24 lines. It is necessary to use these lines as they are not moved up when SCROLL is used.

The speed of the wall moving can be made faster/slower by changing the length of the loop in line 250. You could even introduce another variable which makes the game faster as it goes along.

VARIABLES USED

P = Display file for pokeing the 'V'.
W = Display file for pokeing your score.
S = Score.
H = High Score.
X = General purpose loops
Y = Position for the 'V'.
C = Counter for number of bricks you have hit.
D = Number of bricks to be knocked out.
Z = Part of score being poked.
F = Vertical position for 'V' when being fired at the wall.
A$ = String array for printing "SPLAT" in big letters.
X$ = Score string to be poked on line 24.

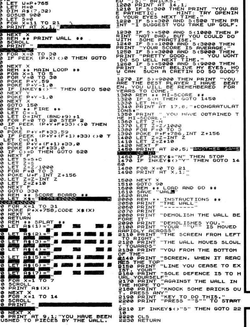
SURROUND

The idea of this game is to trap your opponent, the ZX81, by making it bump into something. The ZX81 will try and trap you, using a rare blend of cunning and intelligence.

Despite this, you'll probably find you know what its next move is going to be after playing it a few times. Then you can get to work, and modify the computer's strategy by changing lines 540 to 570.

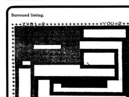
The first player to win five games wins the round.

Wall listings.



```
  20 GOSUB 2000
  30 RAND
  40 LET P=PEEK 16396+PEEK 16397
*256+2
  50 LET W=P+765
  60 LET H=0
  70 DIM A$(7,32)
  80 GOSUB 900
  90 LET S=0
 110 FOR X=11 TO 21
 120 PRINT AT X,1;"██████████████
████████████████"
 130 NEXT X
 150 REM ** PRINT WALL **
 160 SCROLL
 170 PRINT "██████████████████
████████████"
 180 FOR X=0 TO 30
 190 IF PEEK (P+X)<>0 THEN GOTO
1000
 200 NEXT X
 240 REM ** MAIN LOOP **
 250 FOR X=1 TO 5
 260 FOR Y=0 TO 30
 280 POKE P+Y,59
 290 POKE P+Y-1,0
 300 IF INKEY$<>"" THEN GOTO 500
 310 NEXT Y
 320 POKE P+Y-1,0
 330 NEXT X
 340 GOTO 150
 500 REM ** FIRE **
 520 LET C=-1
 530 LET D=INT (RND*9)+1
 540 FOR F=0 TO 20 STEP 2
 550 IF PEEK (P+Y+F*33)<>0 THEN
LET C=C+1
 560 POKE P+Y+F*33,59
 570 IF PEEK (P+Y+(F+1)*33)<>0 T
HEN LET C=C+1
 580 POKE P+Y+(F+1)*33,0
 590 POKE P+Y+F*33,0
 600 IF C>=D THEN GOTO 620
 610 NEXT F
 620 LET S=S+C
 630 LET Z=S
 640 LET Z=Z/1000
 650 FOR F=0 TO 3
 660 POKE W+F,INT Z+156
 670.LET Z=Z-INT Z
 680 LET Z=Z*10
 690 NEXT F
 700 GOTO 330
 900 REM ** SCORE BOARD **
 920 LET X$="SCORE=0000         H
I-SCORE=0000"
 930 FOR X=1 TO 31
 940 POKE P+X+758,CODE X$(X)
 950 NEXT X
 960 RETURN
1000 REM ** SPLAT **
1010 LET A$(1)="██   ██    ██ █
████   ██     █"
1020 LET A$(2)="█ ████ █ ████ █ █
███ ███ ███ ███"
1030 LET A$(3)="█ ████ ████ █ █
███ ███ ███ ███"
1040 LET A$(4)="██   ██    ██ █
███     ███ ███"
1050 LET A$(5)="█████ █ █████ █
███ ███ ███ ███"
1060 LET A$(6)="█ ████ █ █████ █
███ ███ ███ ███"
1070 LET A$(7)="██   ██ █████
 █ ███ ███ ███"
1100 FOR X=1 TO 7
1110 SCROLL
1120 PRINT A$(X)
1130 NEXT X
1140 FOR X=1 TO 14
1150 SCROLL
1160 PRINT "██████████████████
████████████"
1170 NEXT X
1180 PRINT AT 9,1;"YOU HAVE BEEN
CRUSHED TO PIECES BY THE WALL.
   BITS OF YOU ARE ALL OVER THE
PLACE."
1190 PRINT AT 12,1;"YOU DEMOLISH
ED ";S;" BRICKS."
1200 PRINT AT 14,1;
1210 IF S<300 THEN PRINT "YOU AR
E PATHETIC.     HINT: TRY OPENIN
G YOUR EYES NEXT TIME."
1220 IF S>=300 AND S<500 THEN PR
INT "I SUGGEST YOU TAKE UP GOLF.
"
1230 IF S>=500 AND S<1000 THEN P
RINT "NOT BAD, BUT YOU COULD DO
WITH  SOME PRACTISE."
1240 IF S>=1000 AND S<3000 THEN
PRINT "YOUR SCORE IS AVERAGE."
1250 IF S>=3000 AND S<5000 THEN
PRINT "PRETTY GOOD.      YOU WONT
 DO SO WELL NEXT TIME."
1260 IF S>=5000 AND S<9000 THEN
PRINT "I DONT BELIVE MY EYES. HO
W CAN  SUCH A CRETIN DO SO GOOD?
"
1270 IF S>=9000 THEN PRINT "YOU
ARE THE BEST PLAYER I HAVE  BEAT
EN. YOU WILL BE REMEMBERED  FOR
YEARS TO COME."
1300 REM ** HI-SCORE **
1320 IF S<H THEN GOTO 1450
1330 LET H=S
1340 PRINT AT 17,8;"*CONGRATULAT
IONS*"
1350 PRINT " YOU HAVE OBTAINED T
HE HI-SCORE."
1360 LET Z=H
1370 LET Z=Z/1000
1380 FOR F=0 TO 3
1390 POKE P+F+786,INT Z+156
1400 LET Z=Z-INT Z
1410 LET Z=Z*10
1420 NEXT F
1450 PRINT AT 20,5;"ANOTHER GAME
 (Y OR N)"
1460 IF INKEY$="N" THEN STOP
1470 IF INKEY$<>"Y" THEN GOTO 14
60
1480 FOR X=0 TO 21
1490 PRINT AT X,1;"
                              "
1500 NEXT X
1510 GOTO 90
1600 REM ** LOAD AND GO **
1610 SAVE "WALL"
1620 RUN
2000 REM ** INSTRUCTIONS **
2050 PRINT "THE WALL"
2060 PRINT "--------"
2070 PRINT
2080 PRINT "DEMOLISH THE WALL BE
FORE IT"
2090 PRINT "DEMOLISHES YOU."
2100 PRINT "YOUR ""V"" IS MOVED
RAPIDLY ACROSS"
2110 PRINT "THE SCREEN FROM LEFT
 TO RIGHT."
2120 PRINT "THE WALL MOVES SLOWL
Y TOWARDS"
2130 PRINT "YOU FROM THE BOTTOM
OF THE"
2140 PRINT "SCREEN. WHEN IT REAC
HES THE TOP"
2150 PRINT "LINE YOU CEASE TO EX
IST. YOUR"
2160 PRINT "SOLE DEFENCE IS TO H
URL YOURSELF"
2170 PRINT "AGAINST THE WALL IN
THE HOPE TO"
2180 PRINT "KNOCK SOME BRICKS OU
T. PRESS ANY"
2190 PRINT "KEY TO DO THIS."
2200 PRINT "PRESS ""S"" TO START
."
2210 IF INKEY$<>"S" THEN GOTO 22
10
2220 CLS
2230 RETURN
```

Surround listing.

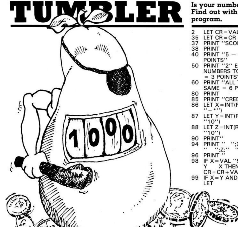


```
  10 CLS
  20 LET Z=0
  30 LET Y=Z
  50 LET A$="5"
  60 LET A=390
  70 LET P=PEEK 16396+PEEK 16397
*256+1
  80 LET B=334

 120 FOR C=6 TO 20
 130 PRINT AT C,31;"++"
 140 NEXT C
 150 FOR C=0 TO 31
 160 PRINT AT 0,C;"+"
 170 PRINT AT 21,C;"+"
 180 NEXT C
 185 LET C=1
 190 PRINT AT 0,3;"ZX81=";Z;TAB
24;"YOU=";Y
 210 LET B=B+C
 220 IF PEEK (P+B)<>0 THEN GOSUB
500
 250 LET B$=A$
 260 LET A$=INKEY$
 270 IF A$<"5" OR A$>"8" THEN LE
T A$=B$
 280 LET A=A+(A$="8")-(A$="5")+(
(A$="6")-(A$="7"))*33
 290 IF PEEK (P+B)<>0 THEN GOTO
1500
 300 POKE P+B,136
 310 IF PEEK (P+A)<>0 THEN GOTO
1000
 320 POKE P+A,128
 390 GOTO 200
 530 LET B=B-C
 540 LET C=33
 550 IF PEEK (B+P+1)=0 THEN LET
C=1
 560 IF PEEK (B+P-1)=0 THEN LET
C=-1
 570 IF PEEK (B+P-33)=0 THEN LET
C=-33
 580 LET B=B+C
 590 RETURN
1050 CLS
1060 PRINT "THE ZX81 WINS YET AG
AIN"
1070 LET Z=Z+1
1075 PAUSE 4E4
1080 IF Z<5 THEN GOTO 40
1090 PRINT "ANOTHER ROUND TO ME"
1100 GOTO 1600
1550 CLS
1560 PRINT "YOU WON, WHAT A RARE
 OCCASION"
1570 LET Y=Y+1
1575 PAUSE 4E4
1580 IF Y<5 THEN GOTO 40
1590 PRINT "A ROUND TO YOU - AMA
ZING"
1600 PRINT "ZX81=";Z,"YOU=";Y
1610 PRINT "ANOTHER ROUND (Y OR
N)"
1620 LET A$=INKEY$
1630 IF A$="Y" THEN RUN
1640 IF A$<>"N" THEN GOTO 1620
```

NUMBER TUMBLER

Is your number up? Find out with this little program.

```
2   LET CR=VAL" 10"
35  LET CR=CR - VAL "1"
37  PRINT "SCORE TABLE"
38  PRINT
40  PRINT "5 - - = 2
    POINTS"
50  PRINT "2" EQUAL
    NUMBERS TOGETHER
    = 3 POINTS"
60  PRINT "ALL THREE THE
    SAME = 6 POINTS"
80  PRINT
85  PRINT "CREDITS = ";CR
86  LET X=INT(RND * VAL
    "-*")
87  LET Y=INT(RND * VAL
    "10")
88  LET Z=INT(RND * VAL
    "10")
90  PRINT"                 "
94  PRINT "   ";X;"   ";Y;
    "   ";Z;"   "
96  PRINT "                "
98  IF X=VAL "5" AND
    Y     X THEN LET
    CR=CR+VAL"2"
99  IF X=Y AND Y=Z THEN
    LET
    CR=CR+VAL"6"
100 IF X=Y AND Z     X OR
    X     Y AND Y=Z THEN
    LET CR=CR+VAL"3"
105 IF CR=PI - PI THEN
    GOTO VAL"130"
107 PRINT "TOUCH ANY
    KEY"
110 PAUSE 4E4
115 CLS
120 GOTO VAL "35"
130 PRINT "FINISHED - NO
    CREDITS LEFT"
```



Written for the unexpanded ZX81, this program is a version of the old favourite 'Fruit machine'. However, in this program, instead of designs, bars, apples, etc, you use numbers.

At the beginning of the game, you have 10 points in credit. These are displayed on a small score table on the screen. When you touch and key, the game starts and three numbers are displayed. If you get any two numbers the same beside each other, you are awarded three points; if you get all three numbers the same you will be awarded the maximum of six points. Should you get the number 5 in the first position, you get a bonus score of two points.

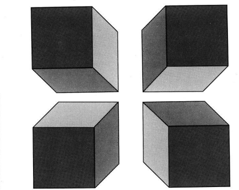
The number of credit points you have decreases by one point for each turn. The game ends when your credit rating reaches zero. As there is limited memory left over, I was unable to include a 'hold' feature, but I'm sure that those of you with extra memory will be able to write a quick routine.

Note that the message 'Score TABLE' in line 37 should be in inverse video for maximum effect.

Luc De Jaeger

3D CUBES

Here is a way to get representations of 3-D cubes on the ZX81. The method is explained and uses a 'selfmade television screen worksheet'.



Here's the way my program works. All you need is a 16K ZX81, a worksheet like the one I've shown here, and a little patience. Although it takes a while to type the program into your ZX81, the results seem pretty good to me.

HOW IT RUNS

Lines 5 to 65 ask for the four co-ordinates of the angular points of the upper surface of the cube. You only have to input the first co-ordinate of the first angular point (E;F), then press NEWLINE (of course), and then input the second co-ordinate, i.e. F . . . . . Mind the right order of the angular points!! Look first of all the television screen worksheet to know where (E,F), (G,H), (I,J),(K,L) are situated, and don't change the order.

Line 70 asks you to give the height of the cube. Don't make M too high otherwise the cube will 'run out of the screen'. A height of about 10 to 25 is recommended. Of course, it all depends upon the position of the chosen co-ordinates of the angular points.

Lines 85 to 180 print the first 'horizontal' edge (see the second example) of the upper surface of the cube. In the first example the first line of the upper surface at the left will be printed.

Lines 185 to 280 print the second 'horizontal' edge.

Lines 285 to 340 print the four vertical edges of the cube.

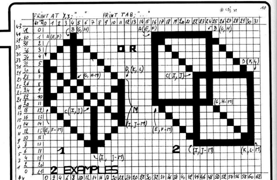
Lines 345 to 440 will print the first horizontal edge of the base (see example two) or the first line in example one.

Lines 445 to 540 dfo the same as lines 345 to 440. Then the second line will be printed.

Lines 545 to 700 print the four oblique lines of the cube.

As you can see the program is rather easy to understand because it's all BASIC.

```
   1 REM **3-DIMENSIONAL CUBES**

   3 REM **GENT - BELGIUM**
   5 PRINT "INPUT ANGULAR POINTS
 OF THE UPPER SURFACE"
   7 PRINT
  10 PRINT "INPUT ANGULAR POINT
A(E,F)"
  15 INPUT E
  20 INPUT F
  25 PRINT "INPUT ANGULAR POINT
B(G,H)"
  30 INPUT G
  35 INPUT H
  40 PRINT "INPUT ANGULAR POINT
C(I,J)"
  45 INPUT I
  50 INPUT J
  55 PRINT "INPUT ANGULAR POINT
D(K,L)"
  60 INPUT K
  65 INPUT L
  70 PRINT "WHAT IS THE HIGHT OF
 THE CUBE ?"
  75 INPUT M
  80 CLS
  85 IF F=H THEN GOTO 95
  90 IF F<>H THEN GOTO 115
  95 FOR N=E TO G
 100 PLOT N,F
 105 NEXT N
 110 GOTO 185
 115 LET O=ABS (F-H)
 120 FOR P=0 TO O-1
 125 IF E<G AND F>H THEN GOTO 15
5
 130 IF E<G AND F<H THEN GOTO 14
5
 135 IF E>G AND F<H THEN GOTO 16
5
 140 IF E>G AND F>H THEN GOTO 17
5
 145 PLOT E+P,F+P
 150 GOTO 180
 155 PLOT E+P,F-P
 160 GOTO 180
 165 PLOT E-P,F+P
 170 GOTO 180
 175 PLOT E-P,F-P
 180 NEXT P
 185 IF J=L THEN GOTO 195
 190 IF J<>L THEN GOTO 215
 195 FOR Q=I TO K
 200 PLOT Q,J
 205 NEXT Q
 210 GOTO 285
 215 LET R=ABS (J-L)
 220>FOR S=0 TO R-1
 225 IF I<K AND J>L THEN GOTO 25
5
 230 IF I<K AND J<L THEN GOTO 24
5
 235 IF I>K AND J<L THEN GOTO 26
5
 240 IF I>K AND J>L THEN GOTO 27
5
 245 PLOT I+S,J+S
 250 GOTO 280
 255 PLOT I+S,J-S
 260 GOTO 280
 265 PLOT I-S,J+S
 270 GOTO 280
 275 PLOT I-S,J-S
 280 NEXT S
 285 FOR T=J-M TO J
 290 PLOT I,T
 295 NEXT T
 300 FOR U=L-M TO L
 305 PLOT K,U
 310 NEXT U
 315 FOR V=F-M TO F
 320 PLOT E,V
 325 NEXT V
 330 FOR W=H-M TO H
 335 PLOT G,W
 340 NEXT W
 345 IF J-M=L-M THEN GOTO 355
 350 IF J-M<>L-M THEN GOTO 375
 355 FOR X=I TO K
 360 PLOT X,J-M
 365 NEXT X
 370 GOTO 445
 375 LET Y=ABS (J-L)
 380 FOR Z=0 TO Y-1
 385 IF I<K AND J-M>L-M THEN GOT
O 415
 390 IF I<K AND J-M<L-M THEN GOT
O 405
 395 IF I>K AND J-M<L-M THEN GOT
O 425
 400 IF I>K AND J-M>L-M THEN GOT
O 435
 405 PLOT I+Z,J-M+Z
 410 GOTO 440
 415 PLOT I+Z,J-M-Z
 420 GOTO 440
 425 PLOT I-Z,J-M+Z
 430 GOTO 440
 435 PLOT I-Z,J-M-Z
 440 NEXT Z
 445 IF F-M=H-M THEN GOTO 455
 450 IF F-M<>H-M THEN GOTO 475
 455 FOR A=E TO G
 460 PLOT A,F-M
 465 NEXT A
 470 GOTO 545
 475 LET B=ABS (F-H)
 480 FOR C=0 TO B-1
 485>IF E<G AND F-M>H-M THEN GOT
O 515
 490 IF E<G AND F-M<H-M THEN GOT
O 505
 495 IF E>G AND F-M<H-M THEN GOT
O 525
 500 IF E>G AND F-M>H-M THEN GOT
O 535
 505 PLOT E+C,F-M+C
 510 GOTO 540
 515 PLOT E+C,F-M-C
 520 GOTO 540
 525 PLOT E-C,F-M+C
 530 GOTO 540
 535 PLOT E-C,F-M-C
 540 NEXT C
 545 LET D=ABS (F-J)
 550 FOR O=0 TO D-1
 555 IF E>I AND F-M>J-M THEN GOT
O 565
 560 IF E<I AND F-M>J-M THEN GOT
O 575
 565 PLOT I+O,J-M+O
 570 GOTO 580
 575 PLOT I-O,J-M+O
 580 NEXT O
 585 LET P=ABS (H-L)
 590 FOR Q=0 TO P-1
 595 IF G>K AND H-M>L-M THEN GOT
O 605
 600 IF G<K AND H-M>L-M THEN GOT
O 615
 605 PLOT K+Q,L-M+Q
 610 GOTO 620
 615 PLOT K-Q,L-M+Q
 620 NEXT Q
 625 LET R=ABS (F-J)
 630 FOR S=0 TO R-1
 635 IF E>I AND F>J THEN GOTO 64
5
 640 IF E<I AND F>J THEN GOTO 65
5
 645 PLOT I+S,J+S
 650 GOTO 660
 655 PLOT I-S,J+S
 660 NEXT S
 665 LET T=ABS (H-L)
 670 FOR U=0 TO T-1
 675 IF G>K AND H>L THEN GOTO 68
5
 680 IF G<K AND H>L THEN GOTO 69
5
 685 PLOT K+U,L+U
 690 GOTO 700
 695 PLOT K-U,L+U
 700 NEXT U
```



Here is the worksheet.

Four of the best on BBC

PROGRAMMING THE BBC MICRO
Edited by Peter Williams

"An excellent review of the potential of the BBC Micro . . . way ahead of its rivals" BBC Micro User

To help you program your BBC Micro. Practical programming points and BBC BASIC as applied to graphics, words, numbers and sound are covered as well as machine code, hex, assembly language, interfacing, file handling and the hardware.

Softcover	*176 pages*	*£6.95*

6502 MACHINE CODE FOR BEGINNERS
A P Stephenson

BASIC is a simple language to learn but it takes up a lot of memory space and is slow to run. Machine code is more complex but it does have considerable benefits and this book shows you how to take advantage of them.

Softcover	*176 pages*	*£5.95*

EXPLOITING BBC BASIC
A P Stephenson and D J Stephenson

A book for BBC Micro users who have mastered the first stages of programming and who now want to exploit their machine to the full.

Two cassettes also available containing some of the longer programs.

Softcover	*176 pages*	*£6.95*

BBC MICRO PROGRAMS IN BASIC
Derrick Daines

Twenty-seven games programs, each with an explanation of how it works and how to play it. They're not only fun, they'll improve your BASIC programming skills at the same time.

Softcover	*128 pages*	*£5.95*

Go and buy your copies now

Newnes Technical Books, Borough Green, Sevenoaks, Kent TN15 8PH

GRAND PRIX DRIVER

Strap yourself in and prepare yourself for the race of your life!

This program, occupying almost 15K, simulates a 10 lap race around a Grand Prix circuit involving all the thrills and spills of the real thing!

Racing around the track clockwise, you move your car using the '5' key to move to the left, the '8' key to move right, the '7' key to move up and the '6' key to move down.

A GRAND TIME

You'll have to watch out for various obstacles on the track, because if you hit them they will cause you to incur time penalties. If you accidentally hit the outer barrier of the track, your car will crash and you are out of the race instantly. Also, if you have an accident in the pits, it is supposed your car will be blown to pieces. Should you crash into the inner barrier, your car will automatically be towed to the pits where it will undergo extensive repairs which will cause you to lose a lap. However, if you manage to crash you car into the barrier again, your car will burst into flames and you are placed out of the race for good!

You begin the race with a

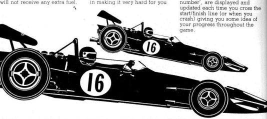
full tank of petrol — 800 litres — which should be enough for about seven to eight laps if you drive carefully. After you have completed seven laps you may enter the pits and be re-fuelled with 50 litres of petrol. You can drive through the pits before this stage of the race, but you will not receive any extra fuel.

When you drive into the pits you must take care as there is usually a lot of oil carelessly strewn around which makes skidding very easy to do.

NO CHEATING

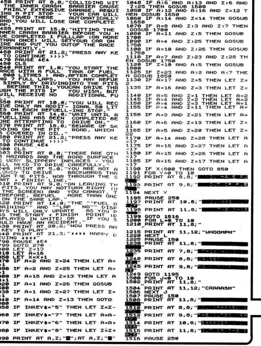
Various routines have been put in making it very hard for you to cheat in this game. You are not allowed to drive through the pits backwards thus re-fuelling more than once a lap, and you may not drive backwards and forwards across the start/finish line.

Various indicators, such as 'fuel remaining' and 'lap number', are displayed and updated each time you cross the start/finish line (or when you crash) giving you some idea of your progress throughout the game.



```
   1 REM "GRAND PRIX DRIVER"
   2 REM BY A D, S L AND R G
         WINDER
  10 REM LIST OF MAIN FEATURES
 100 REM GAME START/SET-UP
         200-299
 105 REM PLAYING INSTRUCTIONS
         300-799
 110 REM PARAMETER SETTING
         3250-3299
 120 REM MAIN DISPLAY 4000-4220
 130 REM VARIABLE RESETTING
         800-849
 140 REM MAIN LOOP 850-1190
 143 REM LAP COUNTER
         2800-2830
 147 REM REFUELLING 3500-3599
 150 REM *MAJOR CRASH ROUTINES*
 152 REM  1500-1515
          (OUTER CRASH BARRIER)
 154 REM  1950-1999
          (ACCIDENT IN PITS)
 156 REM  1200-1249
          (FUEL TANK EXPLOSION)
 160 REM *VARIOUS MINOR CRASH
          SUB-ROUTINES*
 161 REM  1600-1649
          (INNER CRASH BARRIER)
 163 REM  1650-1699
          (START AND FINISH
           LINE BEND)
 164 REM  1700-1749
          (SECOND BEND)
 166 REM  1750-1799 (CHICANE)
 167 REM  1800-1849
          (WAZZERS CURVE)
 168 REM  1850-1899
          (END OF STRAIGHT)
 170 REM VARIOUS INVOLUNTARY

         SKID ROUTINES 1130-1189
 175 REM FUEL EXHAUSTION ROUTINE
         1191-1199
 180 REM NEW LEADER 2500-2799
 185 REM *YOU HAVE WON* ROUTINE
         2000-2499
 190 REM *ANOTHER GAME* ROUTINE
         1533-1575
 200 GOSUB 3250
 210 CLS
```

```
 220 PRINT AT 10,5;"PLEASE ENTER
 YOUR NAME"
 226 PRINT AT 15,6;"THEN PRESS "
"NEW LINE"""
 230 INPUT Y$
 231 CLS
 232 PRINT AT 8,8;"DO YOU REQUIR
E"
 233 PRINT AT 10,5;"PLAYING INST
RUCTIONS?"
 234 INPUT N$
 235 IF N$="YES" THEN GOTO 300
 245 CLS
 250 PRINT AT 12,4;"PRESS ANY KE
Y TO PLAY"
 260 PAUSE 4E4
 270 CLS
 280 LET S=0
 290 LET X=0
 298 GOSUB 4000
 299 GOTO 800
 300 CLS
 310 PRINT AT 1,6;"""GRAND PRIX
DRIVER"""
 320 PRINT AT 2,6;"*************
*****"
 370 PRINT AT 4,0;"THE OBJECT OF
 THE GAME IS TO    DRIVE YOUR CA

R AROUND THE TRACK,UNTIL YOU HAV
E COMPLETED THE    FULL RACE DIS
TANCE, WHICH IS 10 LAPS OF THE C
IRCUIT.  "
 380 PRINT AT 10,0;"THE ""START
+ FINISH POINT"" IS   SITUATED A
T THE BOTTOM CENTRE OFTHE TRACK
AND YOU SHOULD PROCEEDCLOCKWISE.
"

 400 PRINT AT 15,0;"IN ORDER TO
MOVE YOUR CAR       - TO THE LEF
T, PRESS KEY ""5""    - TO THE R
IGHT, PRESS KEY ""8""   - UP THE
 SCREEN, PRESS KEY ""7""  - DOWN
 THE SCREEN, PRESS KEY ""6"""
 410 PRINT AT 21,2;"PRESS ANY KE
Y TO CONTINUE >>>>"
 420 PAUSE 4E4
 430 CLS
 440 PRINT AT 0,0;"IF YOU HIT AN
Y OF THE OBSTACLES,YOU WILL BE P
```



```
ENALISED, USUALLY   BY LOSS OF FI
ME.  HOWEVER, HIT- TING THE OUTE
R BARRIER WILL PUT YOU OUT OF TH
E RACE IMMEDIATELY AND AN ACCIDE
NT IN THE PITS WILLBLOW YOU TO S
MITHEREENS."
 450 PRINT AT 8,0;"COLLIDING WIT
H THE INNER CRASH  BARRIER CAUSE
S FAIRLY EXTENSIVE DAMAGE, REQUI
RING A PIT STOP -  YOUR CAR WILL
 BE TOWED THERE    AUTOMATICALLY
 AND YOU WILL LOSE ONE COMPLETE
LAP."
 455 PRINT AT 15,0;"HITTING THE
INNER CRASH BARRIER BEFORE YOU H
AVE COMPLETED 1 FULLLAP (OR MORE
 THAN ONCE) WILL SETYOUR CAR ON
FIRE AND PUT YOU OUTOF THE RACE
PERMANENTLY."
 460 PRINT AT 21,2;"PRESS ANY KE
Y TO CONTINUE >>>>"
 470 PAUSE 4E4
 480 CLS
 540 PRINT AT 1,0;"YOU START THE
 RACE WITH A FULL  TANK OF FUEL
( 800 LITRES ) AND,AFTER COMPLET
ING 7 FULL LAPS,   YOU MAY REFUE
L SIMPLY BY DRIVINGINTO THE PITS
.  BEFORE THIS, YOUCAN DRIVE THR
OUGH THE PITS IF   YOU WISH, BUT
 WILL RECEIVE NO   ADDITIONAL FU
EL."
 550 PRINT AT 10,0;"YOU WILL REC
EIVE ONLY AN ADDIT- IONAL 50 LIT
RES ON EACH REFUELL-ING STOP."
 560 PRINT AT 14,0;"WAIT UNTIL R
EFUELLING HAS BEEN  COMPLETED BE
FORE ATTEMPTING TO  DRIVE ON."
 570 PRINT AT 18,0;"BEWARE OF SK
IDDING ON THE PIT   ROAD, WHICH
IS COVERED IN OIL."
 580 PRINT AT 21,2;"PRESS ANY KE
Y TO CONTINUE >>>>"
 590 PAUSE 4E4
 600 CLS
 603 PRINT AT 0,0;"THERE ARE OTH
ER HAZARDS AND THE ROAD SURFACE
IS VERY SLIPPERY INPLACES - YOU
WILL NEED TO DRIVE CAREFULLY."
 605 PRINT AT 5,0;"YOU ARE NOT A
LLOWED TO DRIVE     BACKWARDS THR
OUGH THE PITS, NOR THROUGH THE S
TART + FINISH LINE."
 610 PRINT AT 9,0;"ON LEAVING TH
E PITS, YOU MAY NOTTURN RIGHT (U
P THE SCREEN) AND  YOU CANNOT, T
HEREFORE, REFUEL   MORE THAN ONC
E ON THE SAME LAP."
 620 PRINT AT 14,0;"THE ""FUEL R
EMAINING"" AND ""LAP   NO"" DISP
LAYS WILL ONLY UPDATE   AS YOU P
ASS THE START + FINISH  POINT (D
ISPLAYED IN WHITE) OR   IF YOU S
HOULD HAVE AN ACCIDENT."
 630 PRINT AT 20,2;"NOW PRESS AN
Y KEY TO PLAY"
 640 PRINT AT 21,3;"**** HAPPY D
RIVING ****"
 790 PAUSE 4E4
 799 GOTO 270
 800 LET Z=13
 810 LET A=16
 850 LET X=X+1
 870 IF A=2 AND Z=24 THEN LET A=
A+1
 880 IF A=2 AND Z=25 THEN LET A=
A+1
 900 IF A=15 AND Z=13 THEN LET A
=A+1
 920 IF A=1 AND Z=26 THEN GOSUB
3500
 930 IF A=1 AND Z=27 THEN LET Z=
Z+1
 940 IF A=14 AND Z=13 THEN GOTO
2800
 950 IF INKEY$="5" THEN LET Z=Z-
1
 960 IF INKEY$="7" THEN LET A=A-
1
 970 IF INKEY$="6" THEN LET A=A+
1
 980 IF INKEY$="8" THEN LET Z=Z+
1
 990 PRINT AT A,Z;"■";AT A,Z;"■"
1000 IF A=0 THEN GOTO 1500
1010 IF A=19 THEN GOTO 1500
1020 IF Z=0 THEN GOTO 1500
1030 IF Z=31 THEN GOTO 1500
1035 IF A=2 AND Z>25 AND Z<30 TH
EN GOTO 1950
1040 IF A>6 AND A<13 AND Z>6 AND
 Z<25 THEN GOSUB 1600
1050 IF A>12 AND A<16 AND Z=12 T
HEN GOSUB 1650
1060 IF A>14 AND Z=14 THEN GOSUB
 1650
1070 IF A=8 AND Z>3 AND Z<7 THEN
 GOSUB 1700
1080 IF A=11 AND Z<5 THEN GOSUB
1700
1090 IF A=4 AND Z>25 THEN GOSUB
1750
1100 IF A=10 AND Z>26 THEN GOSUB
 1750
1110 IF A=7 AND Z>23 AND Z<28 TH
EN GOSUB 1750
1120 IF Z=10 AND A<5 THEN GOSUB
1800
1125 IF Z=23 AND A>2 AND A<7 THE
N GOSUB 1850
1130 IF A=17 AND Z=5 THEN LET Z=
2
1135 IF A=16 AND Z=3 THEN LET Z=
1
1140 IF A=5 AND Z=1 THEN LET A=2
1145 IF A=4 AND Z=2 THEN LET A=2
1150 IF A=4 AND Z=3 THEN LET A=1
1155 IF A=4 AND Z=11 THEN LET A=
1
1158 IF A=3 AND Z=21 THEN LET A=
5
1160 IF A=6 AND Z=19 THEN LET Z=
21
1165 IF A=5 AND Z=20 THEN LET Z=
22
1170 IF A=14 AND Z=28 THEN LET A
=17
1175 IF A=15 AND Z=27 THEN LET A
=17
1180 IF A=15 AND Z=26 THEN LET A
=17
1185 IF A=15 AND Z=17 THEN LET A
=17
1190 IF X<600 THEN GOTO 850
1191 FOR Y=0 TO 10
1192 PRINT AT 8,8;"████████████████
████"
1193 PRINT AT 9,8;"████████████████
████"
1194 NEXT Y
1195 PAUSE 250
1197 PRINT AT 10,8;"███████████████
████"
1198 PRINT AT 11,8;"███████████████
████"
1199 GOTO 1516
1200 FOR L=0 TO 10
1210 PRINT AT 11,8;"
  "
1215 PRINT AT 11,12;"WHOOMPH"
1220 NEXT L
1225 PAUSE 150
1230 PRINT AT 11,8;"███████████████
████"
1235 PRINT AT 7,8;"████████████████
██"
1240 PRINT AT 8,8;"████████████████
███"
1245 PRINT AT 9,8;"████████████████
██"
1249 GOTO 1195
1500 FOR J=0 TO 10
1502 PRINT AT 11,8;"
  "
1504 PRINT AT 11,12;"CRAAASH"
1506 NEXT J
1507 PAUSE 150
1508 PRINT AT 11,8;"███████████████
████"
1510 PRINT AT 8,8;"████████████████
███"
1512 PRINT AT 9,8;"████████████████
███"
1514 PRINT AT 10,8;"███████████████
███"
1515 PRINT AT 11,8;"███████████████
███"
1516 PAUSE 250
```

```
1520 CLS
1523 IF S>E THEN GOTO 2500
1524 FOR Q=0 TO 10
1525 PRINT AT 5,3;"
 "
1526 PRINT AT 8,9;"
 "
1527 PRINT AT 5,3;"THE CURRENT R
ACE LEADER IS"
1528 PRINT AT 8,9;B$;"( ";E;" LA
PS)"
1532 NEXT Q
1533 PAUSE 150
1535 CLS
1536 PRINT AT 14,8;"ANOTHER GAME
?"
1540 INPUT M$
1542 IF M$="YES" THEN GOTO 210
1544 IF M$="NO" THEN GOTO 1560
1550 PRINT AT 14,0;"U MAY ONLY A
NSWER ""YES"" OR ""NO"""
1552 PAUSE 150
1556 GOTO 1535
1560 CLS
1563 PRINT AT 12,2;"OK. HOPE YOU
 ENJOYED PLAYING"
1566 PRINT AT 14,7;"""GRAND PRIX
 DRIVER"""
1570 PRINT AT 18,9;"BYE BYE FOR
NOW"
1575 STOP
1600 LET S=S-1
1605 IF S<0 THEN GOTO 1200
1610 FOR R=0 TO 10
1611 PRINT AT 11,8;"
 "
1613 PRINT AT 11,12;"GRAUNCH"
1615 NEXT R
1617 PAUSE 150
1620 PRINT AT 11,8;"[illegible]"
1622 PRINT AT 7,8;"[illegible]"
1624 PRINT AT 8,8;"[illegible]"
1626 PRINT AT 9,7;"[illegible]"
1628 PRINT AT 10,8;"[illegible]"
1630 PAUSE 100
1632 PRINT AT 11,7;"[illegible]"
1635 PRINT AT 12,12;"[illegible]"
1640 PAUSE 250
1643 GOSUB 4000
1646 LET A=1
1647 LET Z=30
1648 PRINT AT A,Z;"■";AT A,Z;"■"
1649 RETURN
1650 FOR T=0 TO 10
1652 PRINT AT 11,8;"
 "
1654 PRINT AT 11,12;"ZEEURCH"
1656 NEXT T
1657 PAUSE 150
1658 PRINT AT 11,8;"[illegible]"
1660 PRINT AT 7,1;"[illegible]"
1662 PRINT AT 8,1;"[illegible]"
1663 PRINT AT 9,1;"[illegible]"
1664 PRINT AT 10,1;"[illegible]"
1666 PRINT AT 11,1;"[illegible]"
1668 PAUSE 250
1669 PRINT AT 12,1;"[illegible]"
1670 PRINT AT 13,1;"[illegible]"
1675 LET X=X+40
1676 LET A=16
1678 LET Z=13
1680 PAUSE 250
1690 GOSUB 4000
1695 PRINT AT A,Z;"■";AT A,Z;"■"
1699 RETURN
1700 FOR W=0 TO 10
1702 PRINT AT 11,8;"
 "
1704 PRINT AT 11,11;"SCREEEECH"
1706 NEXT W
```

```
1707 PAUSE 150
1709 PRINT AT 11,8;"[illegible]"
1710 PRINT AT 7,8;"[illegible]"
1712 PRINT AT 8,8;"[illegible]"
1713 PRINT AT 9,8;"[illegible]"
1714 PAUSE 100
1715 PRINT AT 11,7;"[illegible]"
1716 PRINT AT 12,8;"[illegible]"
1720 PAUSE 250
1722 LET A=11
1724 LET Z=5
1725 LET X=X+20
1730 GOSUB 4000
1740 PRINT AT A,Z;"■";AT A,Z;"■"
1749 RETURN
1750 FOR K=0 TO 10
1752 PRINT AT 11,8;"
 "
1754 PRINT AT 11,12;"CRUUNCH"
1756 NEXT K
1758 PRINT AT 11,8;"[illegible]"
1760 PRINT AT 8,8;"[illegible]"
1762 PRINT AT 9,10;"[illegible]"
1764 PAUSE 100
1765 PRINT AT 10,7;"[illegible]"
1766 PRINT AT 11,8;"[illegible]"
1772 LET Z=25
1774 LET A=3
1775 LET X=X+10
1780 PAUSE 250
1790 GOSUB 4000
1795 PRINT AT A,Z;"■";AT A,Z;"■"
1799 RETURN
1800 FOR B=0 TO 10
1802 PRINT AT 11,8;"
 "
1804 PRINT AT 11,11;"SSSPAANGG"
1806 NEXT B
1808 PRINT AT 11,8;"[illegible]"
1810 PRINT AT 7,8;"[illegible]"
1811 PRINT AT 8,8;"[illegible]"
1812 PRINT AT 9,8;"[illegible]"
1814 PAUSE 100
1815 PRINT AT 11,7;"[illegible]"
1816 PRINT AT 12,8;"[illegible]"
1822 LET Z=8
1824 LET A=5
1825 LET X=X+40
1830 PAUSE 250
1840 GOSUB 4000
1845 PRINT AT A,Z;"■";AT A,Z;"■"
1849 RETURN
1850 FOR U=0 TO 10
1852 PRINT AT 11,8;"
 "
1854 PRINT AT 11,12;"SMAAASH"
1856 NEXT U
1857 PAUSE 150
1858 PRINT AT 11,8;"[illegible]"
1860 PRINT AT 7,8;"[illegible]"
1862 PRINT AT 8,8;"[illegible]"
1863 PRINT AT 9,8;"[illegible]"
1864 PRINT AT 10,8;"[illegible]"
1866 PAUSE 250
1867 PRINT AT 11,7;"[illegible]"
1868 PRINT AT 12,8;"[illegible]"
1869 PAUSE 250
1871 LET X=X+60
1872 LET A=2
```

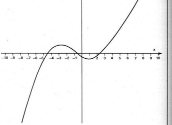
```
1873 LET Z=21
1874 GOSUB 4000
1875 PRINT AT A,Z;"█";AT A,Z;"█"
1899 RETURN
1950 FOR V=0 TO 10
1955 PRINT AT 11,8;"███████████
████"
1960 PRINT AT 11,8;"█████KABOOOM█
████"
1965 NEXT V
1970 PAUSE 150
1978 PRINT AT 11,8;"███████████
████"
1982 PRINT AT 7,8;"████████████
███"
1985 PRINT AT 8,8;"████████████
██"
1990 PRINT AT 9,7;"█████████████
████"
1995 PRINT AT 10,8;"███████████
███"
1999 GOTO 1516
2000 FOR D=1 TO 10
2010 PRINT AT 11,18;"    "
2020 PRINT AT 11,19;S
2030 NEXT D
2040 PAUSE 250
2050 CLS
2100 FOR G=0 TO 53
2103 PRINT "YOU HAVE WON█";
2106 NEXT G
2110 FOR H=0 TO 10
```

```
2113 PRINT AT 1,2;"      "
2116 PRINT AT 17,2;"      "
2120 PRINT AT 2,2;"      "
2123 PRINT AT 18,2;"      "
2126 PRINT AT 3,2;"      "
2130 PRINT AT 19,2;"      "
2133 PRINT AT 4,2;"      "
2136 PRINT AT 20,2;"      "
2140 PRINT AT 5,2;"      "
2143 PRINT AT 21,2;"      "
2150 PRINT AT 1,24;"      "
2160 PRINT AT 17,24;"      "
2170 PRINT AT 2,24;"      "
2180 PRINT AT 18,24;"      "
2190 PRINT AT 3,24;"      "
2200 PRINT AT 19,24;"      "
2210 PRINT AT 4,24;"      "
2220 PRINT AT 20,24;"      "
2230 PRINT AT 5,24;"      "
2240 PRINT AT 21,24;"      "
2251 PRINT AT 1,2;"      "
2261 PRINT AT 17,2;"      "
2271 PRINT AT 2,2;"      "
2281 PRINT AT 18,2;"      "
2291 PRINT AT 3,2;"      "
2301 PRINT AT 19,2;"      "
2311 PRINT AT 4,2;"      "
2321 PRINT AT 20,2;"      "
2331 PRINT AT 5,2;"      "
2341 PRINT AT 21,2;"      "
2350 PRINT AT 1,24;"      "
2360 PRINT AT 17,24;"      "
2370 PRINT AT 2,24;"      "
2380 PRINT AT 18,24;"      "
2390 PRINT AT 3,24;"      "
2400 PRINT AT 19,24;"      "
2410 PRINT AT 4,24;"      "
2420 PRINT AT 20,24;"      "
2430 PRINT AT 5,24;"      "
2440 PRINT AT 21,24;"      "
2450 NEXT H
2460 GOSUB 3250
2470 LET G$="NO ONE"
2499 GOTO 1533
2500 CLS
2510 LET F=E
2520 LET E=S
2530 LET G$=B$
2540 LET B$=Y$
2550 CLS
2560 FOR C=0 TO 20
2570 PRINT AT C,0;"████████████
████████████████████"
2580 NEXT C
2650 FOR P=0 TO 4
2660 PRINT AT 5,11;"██████████"
2670 PRINT AT 6,4;"█████████████
███████████"
2680 PRINT AT 7,9;"████████████
█████"
2690 PRINT AT 9,13;"██████████"
2700 PRINT AT 12,5;"█████████████
██████████████"
2710 PRINT AT 14,13;"████████████
██████"
2720 PRINT AT 17,2;"
            "
2730 PRINT AT 18,2;"
            "
2740 PRINT AT 19,2;"
            "
2745 PRINT AT 5,11;"██████████"
2750 PRINT AT 6,4;"█████████████
███████████"
2755 PRINT AT 7,9;"████████████
███"
2760 PRINT AT 9,13;E;"██████"
2765 PRINT AT 12,5;"█████████████
██████████████"
2770 PRINT AT 14,13;B$
2775 PRINT AT 17,3;"THE DRIVER I
N SECOND PLACE"
2780 PRINT AT 18,9;"IS NOW ";G$
2785 PRINT AT 19,11;"ON ";F;" LA
PS"
2790 NEXT P
2795 PAUSE 300
2799 GOTO 1535
2800 LET S=S+1
2802 PRINT AT 9,11;"    "
2803 PRINT AT 9,11;800-X
2810 PRINT AT 11,19;S
2815 IF S=10 THEN GOTO 2000
2820 LET A=A+1
2830 GOTO 850
3250 LET E=0
3260 LET F=0
3270 LET B$="NO ONE"
3299 RETURN
3500 IF S>6 THEN GOSUB 3550
3520 LET Z=Z+1
3530 PRINT AT A,Z;"█";AT A,Z;"█"
3549 RETURN
3550 LET X=X-50
3560 FOR M=1 TO 25
3570 PRINT AT 1,20;"REFUELLING"
3580 PRINT AT 1,20;"██████████"
3590 NEXT M
3595 PRINT AT 9,11;800-X
3599 RETURN
4000 PRINT AT 0,0;"████████████
████████████████████"
4010 PRINT AT 1,0;"████████████
████████████████████"
4020 PRINT AT 2,0;"████████████
████████████████████"
4030 PRINT AT 3,0;"████████████
████████████████████"
4040 PRINT AT 4,0;"████████████
████████████████████"
4050 PRINT AT 5,0;"████████████
████████████████████"
4060 PRINT AT 6,0;"████████████
████████████████████"
4070 PRINT AT 7,0;"████████████
████████████████████"
4080 PRINT AT 8,0;"████████████
████████████████████"
4090 PRINT AT 9,0;"████████████
████████████████████"
4100 PRINT AT 10,0;"███████████
█████████████████████"
4110 PRINT AT 11,0;"███████████
LAP NO     ███████████"
4120 PRINT AT 12,0;"███████████
█████████████████████"
4130 PRINT AT 13,0;"███████████
█████████████████████"
4140 PRINT AT 14,0;"███████████
█████████████████████"
4150 PRINT AT 15,0;"███████████
█ ███████████████████"
4160 PRINT AT 16,0;"███████████
█████████████████████"
4170 PRINT AT 17,0;"███████████
█████████████████████"
4180 PRINT AT 18,0;"███████████
█████████████████████"
4190 PRINT AT 19,0;"███████████
█████████████████████"
4200 PRINT AT 9,11;800-X
4210 PRINT AT 11,19;S
4220 RETURN
```

John Norbury

EQUATION SOLVER

We show you how to solve equations on the ZX81 using the bisection method.



Suppose that you want to solve an equation of the form f(x) = 0, such as:

$$y = f(x) = x^3 + 3x^2 - 10x - 15$$

The roots of this equation are the values of x for which y = 0. You can find roughly where they occur from a sketch graph (Fig. 1) which shows that the curve cuts the x axis at three places. One is between x = −5 and x = −4, another between x = −1 and x = 0, and the third is between x = 2 and X = 3.

You could find this same information by tabulating the values of y = f(x) for a range of values between, say, −10 and +10 (Fig. 2). Where the sign changes from plus to minus, or vice versa, is the interval in which you could expect to find a root. The table shows that there are again the three intervals (−5, −4), (−1,0) and (2,3) which need to be studied in more detail.

The bisection method takes each interval in turn, divides it into two equal parts, and decides in which half the root is to be found. It then divides that half into two equal parts and repeats the process. This iteration continues until the size of the portion so formed is smaller than the specified error, at which point it gives the mid-point of the portion as the root of the equation. It is here that the graph crosses the x axis and changes sign.

ADVANTAGES AND DISADVANTAGES

Unless you tabulate the function over a very wide range, you cannot be sure of finding all the roots of an equation. Nor can you find any roots that might be complex numbers, such as the solution to the equation $y = x^2 + 1$, for its graph does not even cross the x axis (Fig. 3). However, you can find the solutions to whatever accuracy you like given sufficent iterations and, as the method does not suffer from instability, it always converges to the required answer.

BETWEEN THE LINES

The program will run with about 4½K RAM. The following describes the structure of the program.

Lines 10-110 set up the default values of the parameters, the equation to be solved is held as A$.

Lines 200-360 print the menu and choose the selected subroutine.

Lines 1000-1030 allow the equation to be entered or changed. To prevent program execution stopping with the report code A, powers of x such as x^3 should be entered as x*x*x, and so on.

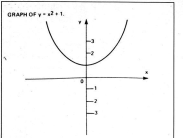
Lines 2000-2080 specify the interval in which you are searching for a root.

Lines 3000-3040 ask for the number values that you want to tabulate.

Lines 4000-4090 tabulate the function over the range set by option 2. If a change of sign occurs between the lowest and highest points of the range, the bisection method can be used. If two or more changes of sign occur, the interval set by option 2 should be reduced to include just one such sign change. if no change of sign occurs, then a wider interval could be tried.

Lines 5000-5040 allow the maximum permitted error in the answer to be set. The smaller this is made, the longer it will take and the more iterations will be needed to reach the solution. In fact the error depends on $1/2^n$, where n is the number of

iterations.
Lines 6000-6230 carry out the actual bisection process, letting the user know if an incorrect interval has been set by option 2.
Lines 7000-7050 list the current values of all parameters, including the equation itself.
Lines 9000-9070 wait for a user to specify either a return to the menu or a printout of screen contents. The program will run as it is even if no printer is available.



HOW TO USE THE PROGRAM

When first run, choose option 1 and enter test equation $x^3 + 3x^2 - 10x - 5$ in the form:

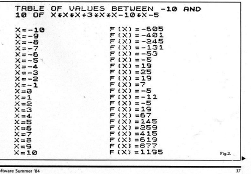
x∗x∗x + 3∗x∗x − 10∗x − 5

Then choose option 2 and specify the interval from − 10 to + 10. Option 3 should be used to ask for 20 tabulation points. Option 4 will list the values of the equation at these 20 points, and will also tell you what to do when the screen fills up. The table should look like that in Fig. 1. Three changes of sign are found, so we call up option 2 again and enter the interval (− 5, − 4).

Next, we choose an error of less than 0.00001 using option 5. Option 6 calculates the root in our chosen interval as − 4.84946. You can then return to option 2 to set the other two intervals in which there is a root of the equation.

Note that the test equation is a cubic, for which there can only be three roots, However, any polynomial or trigonometric equation can be entered, for example, $\sin(x) + \tan(x) - 3x^2 + 4x$.

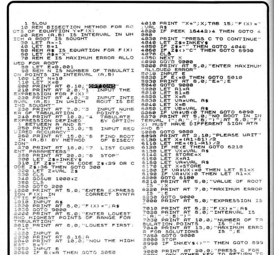
With practice, this becomes an effective and reliable way of solving all sorts of equations — especially recommended for maths homework!



```
TABLE OF VALUES BETWEEN -10 AND
10 OF X*X*X+3*X*X-10*X-5

X=-10                F(X)=-605
X=-9                 F(X)=-401
X=-8                 F(X)=-245
X=-7                 F(X)=-131
X=-6                 F(X)=-53
X=-5                 F(X)=-5
X=-4                 F(X)=19
X=-3                 F(X)=25
X=-2                 F(X)=19
X=-1                 F(X)=7
X=0                  F(X)=-5
X=1                  F(X)=-11
X=2                  F(X)=-5
X=3                  F(X)=19
X=4                  F(X)=67
X=5                  F(X)=145
X=6                  F(X)=259
X=7                  F(X)=415
X=8                  F(X)=619
X=9                  F(X)=877
X=10                 F(X)=1195
```

Fig.2.

$y = f(x) = x^3 + 3x^2 - 10x - 15$



```
   1 SLOW
  10 REM BISECTION METHOD FOR RO
OTS OF EQUATION Y=F(X)
  20 REM (A,B) IS INTERVAL IN WH
ICH A ROOT IS SOUGHT
  30 LET A=-1
  40 LET B=1
  50 REM A$ IS EQUATION FOR F(X)
  60 LET A$="X"
  70 REM E IS MAXIMUM ERROR ALLO
WED FOR ROOT
  80 LET E=0.001
  90 REM N IS NUMBER OF TABULATI
ON POINTS IN INTERVAL (A,B)
 100 LET N=10
 110 LET X=0
 200 PRINT AT 0,10;"OPTIONS"
 210 PRINT AT 2,0;"1  INPUT THE
EXPRESSION FOR F(X)"
 220 PRINT AT 4,0;"2  INPUT INTE
RVAL (A,B) IN WHICH   ROOT IS BE
ING SOUGHT"
 230 PRINT AT 7,0;"3  INPUT NUMB
ER OF TABULATION      POINTS"
 240 PRINT AT 10,0;"4  TABULATE
EXPRESSION DEFINED      BY OPTION
 1 BETWEEN A AND B"
 250 PRINT AT 13,0;"5  INPUT REQ
UIRED ACCURACY"
 260 PRINT AT 15,0;"6  FIND ROOT
 IN (A,B) BY METHOD     OF BISECT
ION"
 270 PRINT AT 18,0;"7  LIST CURR
ENT PARAMETERS"
 280 PRINT AT 20,0;"8  STOP"
 300 LET Z$=INKEY$
 310 IF Z$="" OR CODE Z$<29 OR C
ODE Z$>36 THEN GOTO 300
 320 LET Z=VAL Z$
 330 CLS
 340 GOSUB 1000*Z
 350 CLS
 360 GOTO 200
1000 PRINT AT 5,0;"ENTER EXPRESS
ION F(X) IN           CORRECT SYNTA
CTIC FORM"
1010 INPUT A$
1020 PRINT AT 8,0;"F(X)=";A$
1030 GOTO 9000
2000 PRINT AT 5,0;"ENTER LOWEST
AND HIGHEST POINTS OF RANGE FOR
TABULATION"
2010 PRINT AT 8,0;"LOWEST FIRST:
 A=?"
2020 INPUT A
2030 PRINT AT 8,16;A
2040 PRINT AT 10,0;"NOW THE HIGH
EST: B=?"
2050 INPUT B
2060 IF B<=A THEN GOTO 2050
2070 PRINT AT 10,19;B
2080 GOTO 9000
3000 PRINT AT 5,0;"ENTER NUMBER
OF TABULATION           POINTS"
3010 INPUT N
3020 IF N<>INT N OR N<1 THEN GOT
O 3010
3030 PRINT AT 8,0;"N=";N
3040 GOTO 9000
4000 FOR X=A TO B STEP (B-A)/N
4010 PRINT "X=";X;TAB 15;"F(X)="
;VAL A$
4020 IF PEEK 16442>4 THEN GOTO 4
080
4030 PRINT "PRESS C TO CONTINUE"
4040 LET Z$=INKEY$
4050 IF Z$="" THEN GOTO 4040
4060 IF Z$<>"C" THEN GOTO 8990
4070 CLS
4080 NEXT X
4090 GOTO 9000
5000 PRINT AT 5,0;"ENTER MAXIMUM
 ALLOWED ERROR"
5010 INPUT E
5020 IF E<=0 THEN GOTO 5010
5030 PRINT AT 8,0;"E=";E
5040 GOTO 9000
6000 LET A1=A
6010 LET B1=B
6020 LET X=A
6030 LET VA=VAL A$
6040 LET X=B
6050 LET VB=VAL A$
6060 IF VA*VB<0 THEN GOTO 6090
6070 PRINT AT 5,0;"NO ROOT IN IN
TERVAL (";A;",";B;")";AT 8,0;"F(
X) MUST HAVE DIFFERENT SIGN ATA
AND B"
6080 GOTO 9000
6090 PRINT AT 2,10;"PLEASE WAIT"
6100 LET X=(A1+B1)/2
6110 LET ME=(B1-A1)/2
6120 IF ME<E THEN GOTO 6210
6130 LET VX=VAL A$
6140 LET STORE=X
6150 LET X=A1
6160 LET VA=VAL A$
6170 LET X=STORE
6180 IF VA*VX<=0 THEN LET B1=X
6190 IF VA*VX>0 THEN LET A1=X
6200 GOTO 6100
6210 PRINT AT 5,0;"VALUE OF ROOT
 IS ";X
6220 PRINT AT 7,0;"MAXIMUM ERROR
 IS ";E
6230 GOTO 9000
7000 PRINT AT 5,0;"EXPRESSION IS
"

7010 PRINT AT 6,2;"F(X)=";A$
7020 PRINT AT 8,0;"INTERVAL IS
";A;",";B;")"
7030 PRINT AT 10,0;"NUMBER OF TA
BULATION POINTS        IS ";N
7040 PRINT AT 13,0;"MAXIMUM ERRO
R FOR SOLUTIONS        IS ";E
7050 GOTO 9000
8000 STOP
8990 IF INKEY$<>"" THEN GOTO 899
0
9000 PRINT AT 20,0;"PRESS C FOR
COPY, ANY OTHER KEY TO RETURN TO
 OPTION POINT"
9010 LET Y$=INKEY$
9020 IF Y$="" THEN GOTO 9010
9030 IF Y$<>"C" THEN RETURN
9040 REM 65 SPACES IN NEXT LINE
9050 PRINT AT 20,0;"
                                 "

9060 COPY
9070 GOTO 9000
```

MINEFIELD

You must really watch your step in this program!

All you have to do in this game for your ZX81 is to establsh a safe route through a minefield you know to be planted with 13 deadly mines.

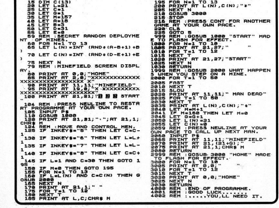
TO THE DEATH

Your mission is of the suicide variety and of the utmost importantance — vital to the war effort in the war you have happened upon. And that is why to a certain degree your lives are expendable. You command three men and their task is to beat a path through the minefield — at least one of your men must get through as three explosions will alert the enemy that someone is trying to cross their defences and your mission will be over.

You move your men across using the cursor keys. No clues or hints are given as to where the mines have been hidden — you must manoeuver your men purely on instinct. Good luck!

Here follows a brief breakdown of the program, line by line:

Lines 5-35	Set up the variables used throughout the program.
Lines 40-55	Provide the geographical parameters of the minefield in which to confine the random deployment of mines.
Lines 60-75	The random deployment of the mines in the minefield.
Lines 80-120	The minefield screen display.
Lines 125-190	Control the movement of the men. (Lines 155-170 compare the men's movements with the 13 mine positions.)
Lines 195-225	Control the end of the game and the re-start if required.
Lines 1000-1030	Contain the code for the word 'START' to flash on the screen at the start of the game.
Lins 2000-2080	Contain the routine for what happens, explosions, etc, when one of your characters steps on one of the hidden mines.
Lines 3000-3030	Contain the code for the word 'HOME' to flash on the screen when you have managed to cross the minefield with one of your characters.



```
   1 REM :SET UP VARIABLES,ETC.
   5 RAND 0
  10 DIM L(13)
  15 DIM C(13)
  20 LET L=21
  25 LET C=1
  30 LET G=1
  35 LET M=157
  40 LET A=18
  45 LET B=3
  50 LET D=31
  55 LET E=0
  59 REM :SECRET RANDOM DEPLOYME
NT  OF MINES.
  60 FOR N=1 TO 13
  65 LET L(N)=INT (RND*(A-B+1)+B
)
  70 LET C(N)=INT (RND*(D-E+1)+E
)
  75 NEXT N
  79 REM :MINEFIELD SCREEN DISPL
AY.
  80 PRINT AT 0,0;"HOME"
  85 PRINT AT 2,0;"XXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX X"
  90 PRINT AT 11,11;"MINEFIELD"
  95 PRINT AT 19,0;"X XXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX"
 100 PRINT AT 21,21;"1 2 3 START
"
 104 REM :PRESS NEWLINE TO RESTA
RT PROGRAMME AT YOUR OWN PACE.
 105 INPUT P$
 110 GOSUB 1000
 120 PRINT AT 21,21;"."; AT 21,1;
CHR$ M
 124 REM :MOVE AND CONTROL MEN.
 125 IF INKEY$="5" THEN LET C=C-
1
 130 IF INKEY$="6" THEN LET L=L+
1
 135 IF INKEY$="7" THEN LET L=L-
1
 140 IF INKEY$="8" THEN LET C=C+
1
 145 IF L=1 AND C=30 THEN GOTO 1
95
 150 IF M=0 THEN GOTO 195
 155 FOR N=1 TO 13
 160 IF L=L(N) AND C=C(N) THEN G
OSUB 2000
 165 NEXT N
 170 PRINT AT 21,1;" "
 175 FOR T=1 TO 12
 180 NEXT T
 185 PRINT AT L,C;CHR$ M
 190 GOTO 125
 194 REM :CLOSE OF GAME AND REST
ART.
 195 FOR N=1 TO 13
 200 PRINT AT L(N),C(N);"*"
 205 NEXT N
 210 GOSUB 3000
 215 STOP
 216 REM :PRESS CONT FOR ANOTHER
GAME AT YOUR OWN PACE.
 220 CLS
 225 GOTO 5
 999 REM :GOSUB 1000 "START" MAD
E TO FLASH FOR EFFECT.
1000 FOR N=1 TO 10
1005 PRINT AT 21,27;"      "
1010 FOR T=1 TO 12
1015 NEXT T
1020 PRINT AT 21,27;"START"
1025 NEXT N
1030 RETURN
1999 REM :GOSUB 2000 WHAT HAPPEN
S WHEN YOU STEP ON A MINE.
2000 FOR T=1 TO 50
2005 FAST
2010 NEXT T
2015 SLOW
2020 PRINT AT 11,11;" MAN DEAD"
2025 FOR T=1 TO 50
2030 NEXT T
2035 PRINT AT L(N),C(N);"*"
2040 LET M=M+1
2041 IF M=160 THEN LET M=0
2045 LET G=G+1
2050 LET L(N)=21
2055 LET C(N)=0
2059 REM :PRESS NEWLINE AT YOUR
OWN PACE TO CALL UP NEXT MAN.
2060 INPUT P$
2065 PRINT AT 11,11;"MINEFIELD"
2070 PRINT AT 21,(21+G);"  "
2075 PRINT AT 21,1;CHR$ M
2080 RETURN
2999 REM :GOSUB 3000 "HOME" MADE
 TO FLASH FOR EFFECT.
3000 FOR N=1 TO 10
3005 PRINT AT 0,0;"    "
3010 FOR T=1 TO 12
3015 NEXT T
3020 PRINT AT 0,0;"HOME"
3025 NEXT N
3030 RETURN
4000 REM :END OF PROGRAMME.
4001 REM :GOOD LUCK.......
4002 REM :.....YOU,LL NEED IT.
```

DEATH RACE

Drive your ZX81 to the limit.

The object of this program is to try and manoeuvre your car (represented by an asterisk) across the screen as many times as possible. The trouble is that your opponents have set up a scheme to try and stop you in your quest by setting up a series of road blocks.

Trouble is that your opponents haven't had much time to set up the roadblocks so some of them are not as good as others. The ineffective roadblocks are represented by grey blocks (made up from the graphics character on the 'H' key) and you get points for smashing through these. But you must take care of the black roadblocks.

To move around the screen, you use the '1' key to move the asterisk up and the 'A' key to move the asterisk down. Points are scored each time you cross the screen. Every fourth time you manage to cross the screen you will be awarded an extra life, but watch out for the black roadblocks as you will lose a life each time you hit one!

TOO EASY?

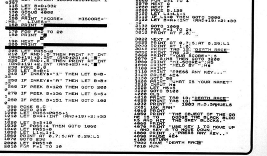
To alter the difficulty of the game, try changing the '.5' in lines 210 and 220. Also, if you want to give yourself more (or less) lives, alter the '5' in line 90. The '4' in line 1030 controls the number of times you have to cross the screen to gain an extra life.

Should you find the game too slow, then try duplicating lines 260, 270 and 280 as lines 8000, 8010 and 8020. Then add the following lines:

260 IF PEEK B< >0 THEN GOTO 8000
8030 GOTO 290

and remove lines 270 and 280.

To SAVE the program, press the Break key, start the tape recorder and type RUN 7000, and press Newline.

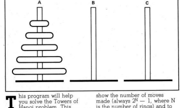


```
   1 REM M.D.SAMUELS
  10 LET H$=""
  20 LET HS=0
  90 LET L1=5
 100 LET S=0
 110 LET A=PEEK 16396+256*PEEK 1
6397
 115 LET B=A+332
 120 LET C=23
 130 GOSUB 4000
 140 CLS
 150 PRINT "SCORE=       HISCORE="
;HS;"  LIVES="
 160 PRINT "█████████████████████
███████████"
 170 FOR F=2 TO 20
 180 PRINT "█                    
         █"
 190 NEXT F
 200 PRINT "█████████████████████
███████████"
 205 LET PASS=0
 210 IF RND>.5 THEN PRINT AT INT
(RND*19)+2,INT (RND*23)+4;"▒"
 220 IF RND>.5 THEN PRINT AT INT
(RND*19)+2,INT (RND*23)+4;"█"
 230 POKE B,0
 235 LET B=B+1
 240 IF INKEY$="1" THEN LET B=B-
33
 250 IF INKEY$="A" THEN LET B=B+
33
 260 IF PEEK B=128 THEN GOTO 200
0
 270 IF PEEK B=136 THEN LET S=S+
1
 280 IF PEEK B=151 THEN GOTO 100
0
 290 POKE B,C
 300 GOTO 210
1000 LET PASS=PASS+1
1010 LET B=A+(INT (RND*19)+2)*33
+2
1020 LET S=S+10
1030 IF PASS<4 THEN GOTO 1060
1040 LET PASS=0
1050 LET L1=L1+1
1060 PRINT AT 0,7;S;AT 0,29;L1
1070 GOTO 230
2000 LET PASS=0
2010 FOR F=1 TO 10
2020 POKE B,128
2030 FOR G=1 TO 4
2040 NEXT G
2050 POKE B,151
2060 FOR G=1 TO 4
2070 NEXT G
2080 NEXT F
2090 POKE B,128
2100 LET L1=L1-1
2110 IF L1=0 THEN GOTO 3000
2120 LET B=A+(INT (RND*19)+2)*33
+2
2130 GOTO 1060
3000 FOR F=1 TO 21
3010 PRINT AT F,0;"             
                  "
3020 NEXT F
3025 PRINT AT 0,7;S;AT 0,29;L1
3030 PRINT AT 1,0;
3040 PRINT TAB 12;"DEATH RACE"
3050 PRINT TAB 12;"▀▀▀▀▀▀▀▀▀▀"
3060 PRINT "YOUR SCORE=";S
3070 IF S>HS THEN GOTO 3200
3080 PRINT "HI-SCORE=";HS
3090 PRINT "HELD BY ";H$
3100 PRINT
3110 PRINT "PRESS ANY KEY..."
3120 PAUSE 4E4
3130 GOTO 80
3200 PRINT "WHAT IS YOUR NAME?"
3210 INPUT H$
3220 LET HS=S
3230 GOTO 3100
4000 CLS
4010 PRINT TAB 13;"DEATH RACE"
4020 PRINT TAB 13;"▀▀▀▀▀▀▀▀▀▀"
4030 PRINT "    1983 M.D.SAMUELS
ZX81 16K RAM"
4040 PRINT
4050 PRINT "THE OBJECT OF THE GA
ME IS TO     DODGE THE BLACK BLOC
KS AND HIT   THE GREY BLOCKS."
4060 PRINT
4070 PRINT "USE KEY 1 TO MOVE UP
, AND KEY A TO MOVE DOWN."
4080 PRINT ,,"PRESS ANY KEY.."
4090 PAUSE 4E4
4100 RETURN
7000 SAVE "DEATH RACE"
7010 RUN
```

G Jackson

TOWERS OF HANOI

The ZX81 solution to this age-old puzzler.



This program will help you solve the Towers of Hanoi problem. This involves a number of rings of different sizes which are stacked on Tower 1 in a similar way to that shown in Fig. 1. The idea of the problem is to eventually stack the rings on Tower 2 or Tower 3 so that at no time does a larger ring rest on a smaller one.

A BIT STEEP?

The driving force of the program is the recursive subroutine (one which calls itself) at lines 50 to 104. The program enters the subroutine with N (the number of rings in play) and breaks out at line 82 with P (the tower number — 1,2 or 3) from which the ring is to move, Q (the number of the 'goto' tower) and N (one less than the number of the moving ring — ring 1 being the smallest). After display of the move, the routine is re-entered at line 84 for the next instruction and then circulates between lines 70 to 104 until the problem is solved. The program finally RETURNs to line 60 to show the number of moves made (always $2^N - 1$, where N is the number of rings) and to wait for key 'P' at line 64.

The recursive subroutine can be modified to run on its own. First, you should add the following line:

```
49 INPUT N
```

where N represents the number of rings. Then, replace lines 60 to 68 with:

```
60 STOP
```

Finally, replace lines 82 to 84 with the following line:

```
82 PRINT "RING";
N+1;"FROM TOWER";P;" TO
TOWER"; Q
```

Then, RUN 49 and enter your value for N.

The action of the routine can be followed through, for N=3 for example, if it is remembered that GOSUB puts its RETURN address (the following line) onto the GOSUB stack and each RETURN removes an address (last in — first out) and goes to it.

START HERE

For newcomers to the ZX81, the following notes may prove of interest.

Lines 40 to 46 produce a number of nine character rings, all PRINTed at column zero; their middle characters from the centre of Tower 1 at column four. The tops of the towers (usually invisible!) are all on screen line five.

At lines 108 to 124, the P and Q values (1,2 or 3) from line 82 are converted to 0, 9 or 18, and are assigned to GF and GT as the PRINT AT column numbers for the 'go from' and 'goto' towers respectively.

The character, R$ (N+1), of the ring to be moved is given by N from line 82, but its line number and the height to which it must be raised are found by PEEKing the towers. This requires the display file reference address, D, which is the central top of Tower 1. (Try adding a line 49 POKE D,23.) The D file addresses of the tops of Towers 2 and 3 are therefore D + 9 and D + 18, so a ring at three lines below the top of Tower 2 will have its centre at addresses D + 9 + 99 and will be found by PEEK (D+T), where T has the value of 108. In that case, the subroutine at lines 166 to 174 will RETURN to lines 114 or 124 with TT=9 and T=108 and assign value 8 to the PRINT AT line number variable X or Y. The difference between eight and three is explained by the tower tops being on screen line five.

The 'other' tower need not be PEEKed; the line number of its upper ring, Z, can be calculated because the total of lines from the tower tops to the upper rings on all three towers is always twice the number of rings in play. Therefore, X — 5 + Y — 5 + Z — 5=2 × N1 (check out lines 34 and 128).

HIGH TIMES

It will be seen that X, Y and Z are inverse measures of the heights of the piles of rings on the 'go from', 'goto' and other tower respectively.

ABS(GF — GT) = 18 means that the 'go from' and 'goto' towers are 18 column apart, so that the other tower is between them. If the height of its pile is greater than that of the GOTO tower, ie Z is less than Y, then the lift (if any) of the moving ring must be determined by Z and not Y — hence lines 134 to 136.

Program with subroutines may suffer from lack of memory if they are frequently interrupted before all the GOSUBs have returned; useless return addresses accumulate in the GOSUB stack, causing it to go down in memory which may be required by the program itself. The effect of this is perpetuated when the program is SAVEd.

In this program, the stack can be cleared in the following way. First add the following line:

```
59 Run 81
```

Then, add the following two lines:

```
61 PRINT AT 14,0;PEEK
   16386+256*PEEK
   16387

81 PRINT AT 14,0;PEEK
   16386+256*PEEK
   16387
```

Finally, change line 82 to read:

```
82 RETURN
```

Repeat the command, RETURN, until the error report shows T/82. Lines 61 and 81 are optional — they show the changes of the addresses of the GOSUB stack (it is 32764 when cleared). Don't forget to remove line 59 and restore line 82 before trying the game again though!

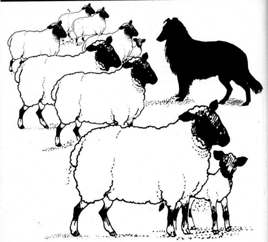
The listing for the program, Towers of Hanoi.

```
  2 PRINT "TOWERS OF HANOI"
  4 DIM R$(9,9)
  6 LET R$(1)="    ▀    "
  8 LET R$(2)="   ▀▀    "
 10 LET R$(3)="   ▀▀▀   "
 12 LET R$(4)="  ▀▀▀▀   "
 14 LET R$(5)="  ▀▀▀▀▀  "
 16 LET R$(6)=" ▀▀▀▀▀▀  "
 18 LET R$(7)=" ▀▀▀▀▀▀▀ "
 20 LET R$(8)="▀▀▀▀▀▀▀▀ "
 22 LET R$(9)="         "
 24 LET M=0
 26 LET D=170+PEEK 16396+256*PE
EK 16397
 28 PRINT "HOW MANY RINGS? (2 T
O 8)"
 30 IF INKEY$<"2" OR INKEY$>"8"
 THEN GOTO 30
 32 LET N=VAL INKEY$
 34 LET N1=N
 36 PRINT AT 1,0;R$(9)+R$(9)+R$
(9)
 38 REM DRAW TOWERS
        ===========
 40 PRINT AT 5,0;
 42 FOR R=1 TO N
 44 PRINT R$(R)
 46 NEXT R
 48 PRINT "    1        2
   3    "
 50 REM "MOVE A RING" ROUTINE
        ======================
 52 LET P=1
 54 LET Q=2
 56 LET R=3
 58 GOSUB 70
 60 PRINT AT 15,0;"OK IN ";M;"
MOVES"
 62 PRINT AT 17,0;"PRESS""P""TO
 GO AGAIN"
 64 IF INKEY$<>"P" THEN GOTO 64
 66 CLS
 68 RUN
 70 IF NOT N THEN RETURN
 72 LET N=N-1
 74 LET Y=Q
 76 LET Q=R
 78 LET R=Y
 80 GOSUB 70
 82 GOTO 108
 84 LET M=M+1
 86 LET Y=P
 88 LET P=R
 90 LET R=Q
 92 LET Q=Y
 94 GOSUB 70
 96 LET Y=P
 98 LET P=Q
100 LET Q=Y
102 LET N=N+1
104 RETURN
106 REM "GO FROM" TOWER
        ===============
108 LET T=P
110 GOSUB 166
112 LET GF=TT
114 LET X=(T-TT)/33+5
116 REM "GO TO" TOWER
        =============
118 LET T=Q
120 GOSUB 166
122 LET GT=TT
124 LET Y=(T-TT)/33+5
126 REM "OTHER" TOWER
        =============
128 LET Z=2*N1-X-Y+15
130 REM DECIDE LIFT
        ===========
132 LET Y1=Y
134 IF ABS (GF-GT)=18 AND Z<Y T
HEN LET Y=Z
136 IF Y>X THEN LET Y=X+1
138 REM LIFT RING
        =========
140 FOR L=X TO Y STEP -1
142 PRINT AT L,GF;R$(9);AT L-1,
GF;R$(N+1)
144 NEXT L
146 REM MOVE TOWER TO TOWER
        ===================
148 FOR C=GF TO GT STEP 2*SGN (
GT-GF)
150 PRINT AT L,C;R$(N+1);AT L,C
R$(9)
152 NEXT C
154 REM DROP RING
        =========
156 FOR L=L TO Y1-1
158 PRINT AT L-1,GT;R$(9);AT L,
GT;R$(N+1)
160 NEXT L
162 GOTO 84
164 REM FIND TOWER TOP AND TOP
RING    =======================
====
166 LET T=9*T-9
168 LET TT=T
170 IF PEEK (D+T) THEN RETURN
172 LET T=T+33
174 GOTO 170
```

Guy Morgan

SHEEPDOG TRIALS

Round up your sheep but don't get penned in!



The object of this game is to round up a number of sheep in the shortest possible time. You are given the choice of how many sheep (between one and six) you would like to round up; one sheep is fairly easy, but if you choose to round up six sheep you will find it very time consuming as they keep wandering out of the pen as others are being rounded up.

You move the sheep by placing the dog near to the sheep using the cursor keys, so that it will move directly away from the dog. However, if you move too close to the sheep, you may find that the sheep will panic and are liable to move in an unpredictable direction.

The dog is allowed three moves between sheep moves; this can be increased if required by changing line 710. The game ends when the dog has rounded up the sheep and is standing guard at the gate of the pen and the dog's three moves have been completed. At any stage of the game, neither the dog or sheep are allowed to jump the fence or wall of the pen.

IT'S A DOG'S LIFE

The dog and sheep are POKEd

onto the screen and the previous positions of the characters are blanked by POKEing with zero. The POKE numbers are calculated as offsets using the number held in the system variable D-FILE as a base point.

The checks on the relationship between sheep and dog are made in terms of x and y co-ordinates, so routines to convert to these are included; they are located at the beginning of the program to speed up the operation. The routine to generate random sheep moves is also placed at the beginning of the program for the same reason. The positions of the sheep in terms of the POKE offsets are held in the array, S(NS).

The timer, which is initialised in lines 707 to 709 and updated by the subroutine at line 2010, makes use of the FRAMES system variable. Lines 9000 and 9010 are the usual ZX81 self-RUN routine.

The listing for Sheepdog Trial.

```
   1 REM ***SHEEPDOG TRIAL***
   9 REM ***RULES***
  10 PRINT AT 0,10;"SHEEPDOG TRI
AL"
  20 PRINT AT 2,0;"YOU MAY CHOOS
E HOW MANY SHEEP TO";"ROUND UP.
NO MORE THAN 6. IF","YOU GET TOO
 CLOSE TO A SHEEP IT ";"WILL PAN
IC AND IF YOU ARE TOO","FAR AWAY
 SHEEP WILL WANDER AT ","RANDOM.
"
  25 PRINT "MOVE DOG WITH THE CU
RSOR KEYS.","TO USE A TURN WITHO
UT MOVING","THE DOG PRESS ANY OT
HER KEY."
  30 PRINT AT 13,0;"PRESS ANY KE
Y TO PLAY."
  40 IF INKEY$="" THEN GOTO 40
  50 CLS
  90 GOTO 300
 100 REM ***CONV. TO CO-ORDS***
 110 LET YS=INT (Z/33)+1
 120 LET XS=(Z-(YS-1)*33)
 130 RETURN
 140 REM ***DOG CO-ORDS***
 150 LET YD=INT (D/33)+1
 160 LET XD=(D-(YD-1)*33)
 170 RETURN
 180 REM ***RANDOM MOVE***
 190 LET XS=XS+1-INT (RND*3)
 200 LET YS=YS+1-INT (RND*3)
 210 GOTO 865
 220 REM ***CONV. TO POKE NO.***
 230 LET WS=XS+33*(YS-1)
 240 RETURN
 300 REM ***NO. OF SHEEP***
 310 PRINT "HOW MANY SHEEP DO YO
U WANT ","TO ROUND UP?"
 315 INPUT NS
 320 IF NS>0 AND NS<7 THEN GOTO
340
 325 PRINT ,,"SORRY, MORE THAN 0
 AND LESS","THAN 7"
 330 GOTO 315
 340 PRINT ,,NS;" IS OKAY."
 345 FOR N=1 TO 100
 346 NEXT N
 350 CLS
 355 REM ***INITIALIZE***
 360 DIM S(NS)
 410 REM ***PEN***
 420 PRINT AT 9,13;"▛▀ ▀▜"
 430 PRINT AT 10,13;"▌  ▐"
 440 PRINT AT 11,13;"▌  ▐"
 450 PRINT AT 12,13;"▙▄▄▟"
 460 REM ***FENCE***
 470 FOR N=0 TO 31
 480 PRINT AT 0,N;".";AT 21,N;".
";
 490 NEXT N
 500 FOR N=0 TO 21
 510 PRINT AT N,0;".";AT N,31;".
";
 520 NEXT N
 530 REM ***GENERATE SHEEP***
 535 LET W=PEEK 16396+256*PEEK 1
6397
 540 FOR N=1 TO NS
 550 LET WS=INT (RND*724)
 560 IF PEEK (W+WS)<>0 THEN GOTO
 550
 570 POKE W+WS,20
 580 LET S(N)=WS
 590 NEXT N
 600 REM ***PLACE DOG***
 610 LET WD=INT (RND*724)
 620 IF PEEK (WD+W)<>0 THEN GOTO
610
 630 POKE WD+W,8
 640 LET D=WD
 700 REM ***START***
 707 LET S=0
 708 POKE 16436,255
 709 POKE 16437,255
 710 FOR N=1 TO 3
 715 LET D1=D
 720 IF INKEY$="" THEN GOTO 720
 730 IF INKEY$="8" THEN LET D=D+
1
 740 IF INKEY$="5" THEN LET D=D-
1
 750 IF INKEY$="6" THEN LET D=D+
33
 760 IF INKEY$="7" THEN LET D=D-
33
 770 IF PEEK (W+D)<>0 THEN LET D
=D1
 780 POKE W+D1,0
 790 POKE W+D,8
 800 NEXT N
 810 REM ***MOVE SHEEP***
 815 GOSUB 140
 820 FOR N=1 TO NS
 830 LET Z=INT (S(N))
 835 GOSUB 100
 840 IF ABS (XS-XD)>3 OR ABS (YS
-YD)>3 THEN GOTO 180
 843 IF ABS (XS-XD)<2 AND ABS (Y
S-YD)<2 THEN GOTO 180
 845 IF XS>XD THEN LET XS=XS+1
 850 IF XS<XD THEN LET XS=XS-1
 855 IF YS>YD THEN LET YS=YS+1
 860 IF YS<YD THEN LET YS=YS-1
 865 GOSUB 220
 870 IF PEEK (W+WS)<>0 THEN GOTO
 900
 875 POKE W+S(N),0
 880 POKE W+WS,20
 885 LET S(N)=WS
 900 NEXT N
 910 REM ***ALL IN PEN?***
 915 LET P=0
 920 FOR N=345 TO 347
 925 IF PEEK (W+N)=20 THEN LET P
=P+1
 930 IF PEEK (W+N+33)=20 THEN LE
T P=P+1
 935 IF P=NS THEN GOTO 1030
 940 NEXT N
 943 GOSUB 2010
 945 GOTO 710
1000 REM ***END OF GAME***
1010 GOSUB 2010
1022 GOTO 1040
1025 REM ***DOG IN GATE?***
1030 IF PEEK (W+313)=8 THEN GOTO
 1000
1035 GOTO 710
1040 LET TM=INT (S/60)
1045 LET TS=INT (S-TM*60)
1050 PRINT AT 18,1;NS;" SHEEP IN
 ";TM;" MIN";
1055 IF TM<>1 THEN PRINT "S";
1060 PRINT AT 18,18;" AND ";TS;"
 SECS"
1070 STOP
2010 LET S=S+(256*(255-PEEK 1643
7)+PEEK 16436)/50
2020 POKE 16437,255
2030 POKE 16436,255
2040 RETURN
9000 SAVE "SHEEPDOG"
9010 GOTO 1
```

David Mold

USER CHARACTER SET

Print upper and lower case characters with your ZX81.

Similar in nature to the user definable graphics program by Chris Callender which appeared in the August/September issue of ***ZX Computing***, this program is rather more flexible in that it will store a character set of up to 99 characters and enable them to be LPRINTed simply by entering a string (X$).

Once you have entered the program, you will be greeted with the prompt:

```
NOW ENTER THE CODES FOR
THE SPECTRUM CHARACTER
SET
```

Simply type in the codes given for the lower case characters (like on the Spectrum), and these will be placed in the array, U.

You will now be given the prompt:

```
ENTER TEXT TO BE LPRINTED
USING NEW CHARACTER SET
```

At this stage, you simply enter what you would like to be printed in upper and lower case.

Any letters you enter as normal video will be printed in lower case. Any letters entered as inverse video will be printed as upper case. Numbers entered as inverse video will actually be printed as normal video numbers, whereas if any numbers are entered as normal video, they will be printed as the extra characters which have been defined for them.

EXTRA, EXTRA

If you have entered your own characters, having answered 'Y' to the prompt:

```
DO YOU WANT TO ENTER
YOUR OWN USER-DEFINED
CHARACTERS? (Y OR N)
```

then they can be obtained by typing in the graphics corresponding to the Sinclair codes 1 to 10 (for details check out Appendix A of the Sinclair ZX81 manual). For example, CHR$ 1 corresponds to the first user-definable graphic.

Character definition should only be attempted if you understand the way that a character is made up from binary numbers (eight for each character). These numbers must then be translated into decimal to be typed in to the program. The extra graphics characters, along with the number that has to be entered to produce them is shown in Fig. 1.

LINE BY LINE

Before typing in or re-LOADing this program from tape, it is important that you enter the following two lines (each followed by Newline) as direct commands:

```
POKE 16389,124
NEW
```

in order to reserve space above RAMtop for the LPRINT routine which is copied out from the ROM in lines 10 to 50.

Here follows a brief breakdown of the structure of the program.

Lines 100 to 170 copy from the ROM into the array S, all the codes for Sinclair's characters, from '0' to 'Z' (normal video).
Lines 200 to 270 copy from the ROM into the array U, all the extra Sinclair symbols, such as '"', ';' etc. These will then be obtainable directly when the program is in use.
Lines 300 to 370 allow for the user to enter the codes for the lower case characters into the array U. (These codes are shown in Fig. 1). Every other line of this printout has the codes to be entered. separated by '/' to represent Newline, and above each set of codes is the number that should appear at the top of the screen when these codes are to be entered (and represent the codes' positions in the array U).
Lines 500 to 700 allow you to enter wherever you wish to be printed using the new character

set, and then looks up the codes for these characters in the appropriate arrays, mapping them into the array A, for LPRINTing using the subroutine from Sinclair's printer manual (lines 9990 to 9999). When all the arrays have been entered, lines 100 to 500 can be deleted. Line 1 should then be changed to read 1 SAVE "Characters". Connect up your cassette, start it recording, then enter GOTO 1 as a direct command. **Do not use RUN** or the arrays will be wiped. Make sure that all DIM statements have been erased. The program, will begin straight away, which avoids the danger of the user typing RUN instead of GOTO to begin.

```
28
255/0/0/0/0/0/0/0/
29
0/16/16/16/16/0/16/0/
30
60/66/153/165/165/158/64/63/
31
0/36/126/36/36/126/36/0/
32
16/56/88/146/16/16/16/16/
33
0/98/100/8/16/38/70/0/
34
0/48/72/50/76/72/54/0/
35
0/8/8/16/0/0/0/0/
36
0/126/36/36/36/36/36/0/
37
60/66/153/161/161/153/66/60/
38
0/0/60/2/62/66/62/0/
39
0/64/64/124/66/66/124/0/
40
0/0/60/64/64/64/60/0/
41
0/2/2/62/66/66/62/0/
42
0/0/60/66/124/64/62/0/
43
0/12/16/56/16/16/16/0/
44
0/0/60/66/66/62/2/60/
45
0/64/64/120/68/68/68/0/
46
0/16/0/48/16/16/56/0/
47
0/4/0/12/4/4/68/56/
48
0/32/40/48/48/40/36/0/
49
0/32/32/32/32/32/24/0/
50
0/0/104/84/84/84/84/0/
51
0/0/120/68/68/68/68/0/
52
0/0/56/68/68/68/56/0/
53
0/0/120/68/68/120/64/64/
54
0/0/60/68/68/60/4/4/
55
0/0/28/32/32/32/32/0/
56
0/0/60/64/56/4/120/0/
57
0/32/112/32/32/32/24/0/
58
0/0/68/68/68/68/56/0/
59
0/0/68/68/40/40/16/0/
60
```

Fig. 1. The codes for the lower case characters — to be entered into the array. u.

```
   1 SAVE "CHARACTERS"
   5 FAST
  10 FOR I=0 TO 112
  20 POKE 31744+I,PEEK (2161+I)
  30 NEXT I
  40 POKE 31800,63
  50 POKE 31857,201
 100 DIM S(36,8)
 110 LET X=7903
 120 FOR N=1 TO 36
 130 FOR P=1 TO 8
 140 LET S(N,P)=PEEK (X+P)
 150 NEXT P
 160 LET X=X+8
 170 NEXT N
 200 DIM U(63,8)
 205 LET X=7767
 210 FOR N=11 TO 27
 220 FOR P=1 TO 8
 230 LET U(N,P)=PEEK (X+P)
 240 NEXT P
 250 LET X=X+8
 260 NEXT N
 270 GOTO 500
 300 PRINT AT 0,0;">NOW ENTER TH
E CODES FOR THE     SPECTRUM CHAR
ACTER SET"
 310 FOR N=28 TO 63
 320 PRINT AT 3,0;N
 330 FOR P=1 TO 8
 340 PRINT AT 4,0;P
 350 INPUT U(N,P)
 360 NEXT P
 370 NEXT N
 400 CLS
 410 PRINT ">DO YOU WANT TO ENTE
R YOUR OWN  USER-DEFINED CHARACT
ER? (Y OR N)"
 420 INPUT A$
 425 IF A$="N" THEN GOTO 500
 430 FOR N=1 TO 10
 440 PRINT AT 3,0;N
 450 FOR P=1 TO 8
 460 PRINT AT 4,0;P
 465 INPUT U(N,P)
 470 NEXT P
 480 NEXT N
 500 CLS
 505 SLOW
 510 PRINT "ENTER TEXT TO BE LPR
INTED USING NEW CHARACTER SET:"
 520 INPUT X$
 522 FAST
 525 DIM A(32,8)
 530 FOR X=1 TO LEN X$
 540 IF X$(X)=" " THEN GOTO 700
 550 IF CODE X$(X)>127 THEN GOTO
 600
 560 FOR Y=1 TO 8
 570 LET A(X,Y)=U(CODE X$(X),Y)
 580 NEXT Y
 590 GOTO 700
 600 FOR Y=1 TO 8
 610 LET A(X,Y)=S(CODE X$(X)-155
Y)
 620 NEXT Y
 700 NEXT X
9990 FOR J=1 TO 32
9991 FOR K=1 TO 8
9992 POKE 32255+K+8*(J-1),A(J,K)
9993 NEXT K
9994 NEXT J
9995 FOR H=0 TO 31
9996 POKE 16444+H,H
9997 NEXT H
9998 LET HPRINT=USR 31744
9999 GOTO 500
```

Paul Toland. Dilwyn Jones

THE SHARP END

Execution and Anagrams are two great word games from Paul for the 16K ZX81, but if you have only 1K try Crasher from Dilwyn.

Execution is a graphic variation on the hangman game. The computer chooses a word randomly from its vocabulary, and you have to guess what the word is, by entering letters one at a time.

You're allowed 10 incorrect guesses before the blade on the guillotine does its job. There is no need to press NEWLINE when entering guesses.

Before you run the program the first time, enter DIM D$(30, 10) as a direct command, then start it with GOTO 1 (not RUN). Then, you'll be asked to enter 30 words, one at a time. For subsequent runs, start with GOTO 5, and your vocabulary will be safe. Enter "N" if you do not want another game at the end. SAVE the program by entering GOTO 390. Note that the inverse E in line 390 should be entered as an ordinary E.

ANAGRAMS

The ZX81 randomly selects and scrambles a word. You must repeatedly enter what you think is the correct word, and the computer will underline those letters which are in the correct position.

You have a maximum number of attempts before the computer stops you and reveals the word. You use "N" if you don't want another game, or just press NEWLINE/ENTER if you do.

Entering the initial vocabulary, and the subsequent changing of it, is the same as in Execution. Save the program by entering GOTO 340, and restart the program without losing your vocabulary, by entering GOTO 10. You'll need to set up an array, by entering DIM D$(30, 10) in the direct mode before you begin, then start with GOTO 1.

It takes some time to scramble a word, so don't be surprised if nothing happens for the first few seconds of the game.

CRASHER

Nothing to do with some computers' habits of blanking out! You are drifting in space, the object being to clear up as much debris as you can, particularly those with a high CODE value. The debris is valued according to the code of the character of debris. The letters score highest, then numbers and so on. Ignore inverse characters if they appear for they do not score because they've been affected by solar radiation and mutated to their inverse form, so are not worth anything. The keys 5 and 8 steer you left and right in the direction of the arrows on the keys. You score when the debris is directly ahead of you on screen so you 'crash' into it. The position of your spaceship on screen is set by X (horizontal position) and Y (vertical position). They are initially set by lines 10 and 20 to a position at about the middle of the playing area of the screen. The variable S records your score and is set to 0 at the start of the game by line 30. F is the variable that controls the duration of the game — it does not count in any particular unit but is convenient way of controlling the length of the game. The time is clocked up in line 50, one unit of space-time at a time(!) The statement LET F=F+ — may confuse you a little since it is hard to imagine F being equal to F+1.

TRUTH

It actually means add 1 to the old value of F to make the new value. Line 60 is rather complicated. It determines where the character is PRINTed, which character is PRINTed and whether it's an inverse character. Let us look at the position first. The character is placed at the bottom of the playing area (the Y co-ordinate is set at 20 and C co-ordinate to a value from 0 to 19 by the random number expression. The expression after the semi-colon generates a character at random from the number generated in brackets after CHR$. The number generated is a random number from 1 to 63. The following expression may look rather strange, but all it does is determine whether to add 0 or 128 to this number (ie determine whether the character generated is inverse or not. It is a special use of the function AND. What it does is look at the following expression, add 0 to the value if it is not true, or the value before AND if it is true, so that 128 is added to the random number only if RND is less than .2. This makes it roughly a one in five chance of the character being an inverse, non-scoring character. The statement after this erases the old position of the V before it is scrolled up the screen by line

70. This ensures the V is not PRINTed anywhere other than the middle of the screen. Line 80 checks the keyboard to see whether you're steering to the left, right, or keeping it stationary, and changes the value of X accordingly.

PRINTING AND RUSHING

The V is PRINTed in its new position in line 90. You can see how the effect given is that the spaceship (V) remains stationary while space seems to rush past. The second part of line 90 moves the PRINT position to immediately ahead of the position of the spaceship. Therefore we may find what is immediately in front of the spaceship by means of the line 100, which finds the CODE of the character stored at that memory location by PEEKing the address held in the system variables 16398 and 16399. Line 110 checks to see if this character ahead of the spaceship by means of the line (ie it has a CODE of less than 64 rather than less than 128 as you might expect from an inverse character detector). This would permit CHR$ 118 NEWLINE markers that SCROLL might push up the screen. You may like to add this line which will stop the program if you hit a radiation-mutated piece of debris:

115 IF P >127 THEN STOP

EXECUTION LISTING

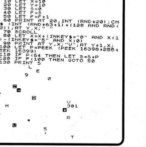
```
  1 FOR I=1 TO 30
  2 INPUT D$(I)
  3 NEXT I
  5 PRINT AT 2,10;"EXECUTION"
  7 PRINT AT 0,0;
 10 RAND
 20 LET G=0
 30 LET C=G
 40 FOR I=1 TO 10
 50 PRINT I;TAB 2;"▌   ▌"
 60 NEXT I
 70 PRINT "  █████ "
 80 PRINT AT 8,3;"O O"
 90 PRINT AT 9,3;">=<"
100 LET R=INT (RND*30)+1
110 LET W$=""
120 FOR I=1 TO 10
130 IF D$(R,I)>" " THEN LET W$=
W$+D$(R,I)
140 NEXT I
150 LET L=LEN W$
160 PRINT AT 8,20;"THE WORD"
165 LET C$="???????????"( TO L)
170 PRINT AT 10,20;C$;AT 21,5;"
TYPE A LETTER"
180 PRINT AT G,3;"████"
190 IF INKEY$<"A" OR INKEY$>"Z"
 THEN GOTO 190
195 LET G$=INKEY$
200 FOR I=1 TO L
210 IF G$<>W$(I) THEN GOTO 240
220 LET G=G-1*(G>=0)
230 LET C$(I)=W$(I)
240 NEXT I
250 IF C$=W$ THEN GOTO 320
260 PRINT AT G,3;"    "
270 LET G=G+1
280 IF G<10 THEN GOTO 170
290 PRINT AT 12,3;"+ +"
300 PRINT "   -"
310 PRINT AT 2,20;"COMPLETED"
315 GOTO 330
320 PRINT AT 2,20;"SUSPENDED"
330 PRINT AT 10,20;W$
340 PRINT AT 21,5;"TRY AGAIN?
"
350 INPUT A$
360 IF A$="N" THEN STOP
370 CLS
380 GOTO 5
390 SAVE "EXE█"
400 GOTO 5
```

ANAGRAMS LISTING

```
  1 FOR I=1 TO 30
  2 INPUT D$(I)
  3 NEXT I
 10 RAND
 20 PRINT "       ANAGRAMS"
 30 LET R=INT (RND*30)+1
 40 LET W$=""
 50 FOR I=1 TO 10
 60 IF D$(R,I)>" " THEN LET W$
W$+D$(R,I)
 70 NEXT I
 80 LET L=LEN W$
 90 LET S$="            "( TO L)
100 FOR I=1 TO L
110 LET R=INT (RND*L)+1
120 IF S$(R)>" " THEN GOTO 110
130 LET S$(R)=W$(I)
140 NEXT I
150 PRINT "YOUR ANAGRAM IS ";S█
160 FOR J=1 TO 9
170 PRINT J;"  ";
180 INPUT G$
190 PRINT G$
200 IF G$=W$ THEN GOTO 280
205 PRINT " ";
210 FOR I=1 TO LEN G$
215 IF I>L THEN GOTO 245
220 IF G$(I)=W$(I) THEN PRINT "
-";
230 IF G$(I)<>W$(I) THEN PRINT
" ";
240 NEXT I
245 PRINT
250 NEXT J
260 PRINT "TIME UP - THE WORD I
S ";W$
270 GOTO 290
280 PRINT "THAT""S IT - ";
290 PRINT "TRY AGAIN ?"
300 INPUT A$
310 IF A$="N" THEN STOP
320 CLS
330 GOTO 10
340 SAVE "ANA█"
350 GOTO 10
```

CRASHER LISTING

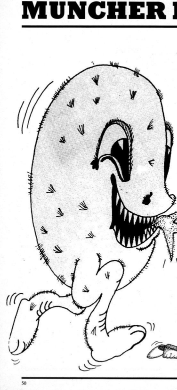
```
 10 LET X=10
 20 LET Y=10
 30 LET S=0
 40 LET F=5
 50 LET F=F+1
 60 PRINT AT 20,INT (RND*20);CH
R$ (INT (RND*63+1)+(128 AND RND<
.2));AT Y,X;" "
 70 SCROLL
 80 LET X=X+(INKEY$="8" AND X<1
9)-(INKEY$="5" AND X>0)
 90 PRINT AT Y,X;"V";AT Y+1,X;
100 LET P=PEEK (PEEK 16398+256*
PEEK 16399)
110 IF P<64 THEN LET S=S+P
120 IF F<100 THEN GOTO 50
130 PRINT S
```



Robert Turner

MUNCHER II

Monsters will try to catch you while you're quietly going dotty!

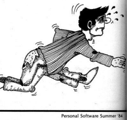


In the June/July issue of ***ZX Computing,*** 1983, appeared the program Muncher — a Spectrum version of the popular arcade game of the dot-eating kind. For all those of you with 16K ZX81s the author, Robert Turner, has converted the program for you.

For those not familiar with this 'infamous' game, you control a small character, an arrow which travels around a small maze eating dots as it goes. That's not quite the end of the story though! As in all good arcade games there should be omnipresent monsters — and this game is no exception. You are chased around the maze by two monsters eager to halt your progress.

However, all is not lost as there are power pills situated in the corners of the maze which, when consumed, give your character the ability to destroy the monsters. You'll know when the monsters are vulnerable as they develop 'eyes'. When you have eaten a power pill, don't hang around as the monsters try to escape.

You score one point each

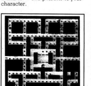


time you eat a dot, and power pills and monsters are worth 10 points. Every now and then a '£' sign will appear under the monster's den and this too is worth 10 points if consumed.

VARIABLES

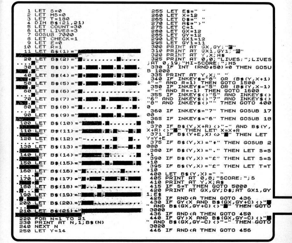
The variables used in the program, Muncher II, are as follows:

S — Score
HS — High Score
T — The score when the screen has been cleared.
B$ — The variable which stores the maze.
COUNT — If this variable is less than 30 the monsters can be eaten. If it is more than 30 the monsters can eat your character.
LIVES — The number of lives you have left.
Y and X — The position of your character.



A$ — The shape of your character.
GX and GY — The position of the first monster.
GX1 and GY1 — The position of the second monster.
C — This variable controls the movement of the monster. If C = −1 then the monster moves away from your character, if C = 1 then the monster chases you.
D$ — The shape under the first monster.
E$ — The shape under the second monster.
A — The skill level.
N — Allotted for general use.

So, don't just sit there — type in the program and watch out for the monsters!

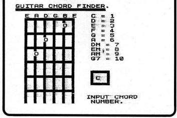


```
  1 LET S=0
  2 LET HS=0
  3 LET T=180
  4 DIM B$(21,21)
  5 LET COUNT=30
  6 LET LIVES=3
  7 GOSUB 7000
  8 LET CHECK=1
  9 LET E=0
 10 LET R=1
 11 LET B$(1)="█████████████████████"
 20 LET B$(2)="█.........█.........█"
 30 LET B$(3)="█.███.███.█.███.███.█"
 40 LET B$(4)="█*███.███.█.███.███*█"
 50 LET B$(5)="█.███.███.█.███.███.█"
 60 LET B$(6)="█...................█"
 70 LET B$(7)="█.███.█.█████.█.███.█"
 80 LET B$(8)="█.....█...█...█.....█"
 90 LET B$(9)="█████.███.█.███.█████"
100 LET B$(10)="█████.█.......█.█████"
110 LET B$(11)="█████.█.█---█.█.█████"
120 LET B$(12)="-.......█   █.......-"
130 LET B$(13)="█████.█.█---█.█.█████"
140 LET B$(14)="█████.█.......█.█████"
150 LET B$(15)="█████.█.█████.█.█████"
160 LET B$(16)="█.........█.........█"
170 LET B$(17)="█*███.███.█.███.███*█"
180 LET B$(18)="█...█...........█...█"
190 LET B$(19)="███.█.█████████.█.███"
200 LET B$(20)="███...............███"
210 LET B$(21)="█████████████████████"
220 FOR N=1 TO 21
230 PRINT AT N,1;B$(N)
240 NEXT N
250 LET Y=14
255 LET E$=" "
260 LET X=12
265 LET D$=" "
270 LET A$=">"
275 LET C=1
280 LET GX=12
285 LET GY=12
290 LET GX1=12
295 LET GY1=11
300 PRINT AT GX,GY;"█"
310 PRINT AT GX1,GY1;"█"
320 PRINT AT Y,X;A$
325 PRINT AT 0,0;"LIVES:";LIVES
;AT 0,19;"HI-SCORE:";HS
330 IF INT (RND*50)=0 THEN GOSU
B 1000
335 PRINT AT Y,X;" "
340 IF INKEY$="8" OR (B$(Y,X+1)
="-" AND R=1) THEN GOTO 1500
350 IF INKEY$="5" OR (B$(Y,X-1)
="-" AND R=-1) THEN GOTO 1600
355 IF INKEY$<>"5" AND INKEY$<>
"6" AND INKEY$<>"7" AND INKEY$<>
"8" AND INKEY$<>"" THEN GOTO 400
0
360 IF INKEY$="7" THEN GOSUB 17
00
365 IF INKEY$="6" THEN GOSUB 18
00
370 IF B$(Y,X+R)<>"-" AND B$(Y,
X+R)<>"█" THEN LET X=X+R
371 IF B$(Y+E,X)<>"█" THEN LET
Y=Y+E
375 IF B$(Y,X)="*" THEN GOSUB 2
000
380 IF B$(Y,X)="." THEN LET S=S
+1
390 IF B$(Y,X)="£" THEN LET S=S
+10
395 IF B$(Y,X)="£" THEN LET T=T
+10
400 LET B$(Y,X)=" "
405 PRINT AT 0,8;"SCORE:";S
410 PRINT AT Y,X;A$
415 IF S=T THEN GOTO 5000
420 PRINT AT GX,GY;D$;AT GX1,GY
1;E$
425 IF RND<A THEN GOTO 436
430 IF GY<X AND B$(GX,GY+C)<>"█
" AND B$(GX,GY+C)<>"█" THEN GOTO
3000
436 IF RND<A THEN GOTO 450
440 IF GY>X AND B$(GX,GY-C)<>"█
" AND B$(GX,GY-C)<>"-" THEN GOTO
3020
446 IF RND<A THEN GOTO 456
```

```
 450 IF GX<Y AND B$(GX+C,GY)<>"■
" THEN GOTO 3040
 455 IF RND<A THEN GOTO 470
 460 IF GX>Y AND B$(GX-C,GY)<>"■
" THEN LET GX=GX-C
 465 IF RND<A THEN GOTO 474
 470 IF GY1<X AND B$(GX1,GY1+C)<
>"■" AND B$(GX1,GY1+C)<>"-" THEN
 GOTO 3060
 474 IF RND<A THEN GOTO 478
 475 IF GY1>X AND B$(GX1,GY1-C)<
>"■" AND B$(GX1,GY1-C)<>"-" THEN
 GOTO 3080
 478 IF RND<A THEN GOTO 486
 480 IF GX1<Y AND B$(GX1+C,GY1)<
>"■" THEN GOTO 3100
 486 IF RND<A THEN GOTO 500
 490 IF GX1>Y AND B$(GX1-C,GY1)<
>"■" THEN LET GX1=GX1-C
 500 LET COUNT=COUNT+1
 510 IF COUNT>=30 THEN LET C=1
 520 IF C=-1 THEN PRINT AT GX,GY
;"""""
 525 IF C=1 THEN PRINT AT GX,GY;
"■"
 530 IF C=-1 THEN PRINT AT GX1,G
Y1;"""""
 535 IF C=1 THEN PRINT AT GX1,GY
1;"■"
 540 IF GX=Y AND GY=X THEN GOTO
9000
 550 IF GX1=Y AND GY1=X THEN GOT
O 9000
 560 LET D$=B$(GX,GY)
 570 LET E$=B$(GX1,GY1)
 580 LET CHECK=1
 600 GOTO 330
1000 IF B$(14,11)="£" THEN RETUR
N
1010 PRINT AT 14,11;"£"
1020 IF B$(14,11)="." THEN LET T
=T-1
1030 LET B$(14,11)="£"
1040 RETURN
1500 PRINT AT Y,X;" "
1510 LET A$=">"
1520 IF B$(Y,X+1)="-" THEN LET C
HECK=0
1530 IF CHECK=0 THEN LET X=2
1535 IF CHECK=0 THEN GOTO 375
1540 IF B$(Y,X+1)<>"■" THEN LET
R=1
1545 IF B$(Y,X+1)<>"■" THEN LET
E=0
1550 GOTO 370
1600 PRINT AT Y,X;" "
1610 LET A$="<"
1620 IF B$(Y,X-1)="-" THEN LET C
HECK=0
1630 IF CHECK=0 THEN LET X=20
1635 IF CHECK=0 THEN GOTO 375
1640 IF B$(Y,X-1)<>"■" THEN LET
R=-1
1645 IF B$(Y,X-1)<>"■" THEN LET
E=0
1650 GOTO 370
1700 PRINT AT Y,X;" "
1710 IF B$(Y-1,X)<>"■" AND B$(Y-
1,X)<>"-" THEN LET E=-1
1715 IF B$(Y-1,X)<>"■" AND B$(Y-
1,X)<>"-" THEN LET R=0
1720 LET A$="A"
1730 RETURN
1800 PRINT AT Y,X;" "
1810 IF B$(Y+1,X)<>"■" AND B$(Y-
1,X)<>"-" THEN LET E=1
1815 IF B$(Y+1,X)<>"■" AND B$(Y-
1,X)<>"-" THEN LET R=0
1820 LET A$="V"
1830 RETURN
2000 LET S=S+10
2010 LET C=-1
2020 LET B$(Y,X)=" "
2030 PRINT AT Y,X;A$
2040 LET T=T+10
2050 LET COUNT=0
2060 RETURN
3000 LET GY=GY+C
3010 GOTO 470
3020 LET GY=GY-C
3030 GOTO 470
3040 LET GX=GX+C
3050 GOTO 470
3060 LET GY1=GY1+C
3070 GOTO 500
3080 LET GY1=GY1-C
3090 GOTO 500
3100 LET GX1=GX1+C
3110 GOTO 500
4000 LET R=0
4010 LET E=0
4020 GOTO 360
5000 CLS
5005 PRINT AT 10,31;"*"
5010 FOR N=1 TO 27
5020 PRINT AT 10,N;" ■  >"
5025 REM SHEET CLEARED
5030 NEXT N
5040 FOR N=27 TO 1 STEP -1
5050 PRINT AT 10,N;"""   < "
5060 NEXT N
5070 LET T=T+180
5080 GOTO 8
7000 PRINT AT 0,12;"MUNCHER II";
TAB 11;"▀▀▀▀▀▀▀▀▀▀"
7010 PRINT AT 9,0;"-------------
-------------------"
7020 PRINT AT 11,3;"SELECT SSILL
 LEVEL (1-5)"
7030 PRINT TAB 7;"(5 IS THE EASI
EST)"
7040 PRINT AT 14,0;"-------------
-------------------"
7050 PRINT TAB 5;"    ROBERT TUR
NER 1983"
7060 LET A=CODE INKEY$-28
7070 IF A>5 OR A<1 THEN GOTO 706
0
7080 LET A=A/10
7090 LET A=A+.1
7100 CLS
7110 RETURN
8000 IF GY<>X OR GX<>Y THEN GOTO
 8060
8010 LET S=S+10
8020 LET T=T+10
8030 LET GY=12
8040 LET GX=12
8045 PRINT AT Y,X;A$
8050 GOTO 500
8060 IF GY1<>X OR GX1<>Y THEN GO
TO 500
8070 LET S=S+10
8080 LET T=T+10
8090 LET GY1=11
8100 LET GX1=12
8105 PRINT AT Y,X;A$
8110 GOTO 500
9000 IF COUNT<=30 THEN GOTO 8000
9005 LET LIVES=LIVES-1
9010 FOR N=1 TO 10
9015 PRINT AT Y,X;A$
9016 FOR Q=1 TO 2
9017 NEXT Q
9018 PRINT AT Y,X;CHR$ (CODE A$+
128)
9019 NEXT N
9020 IF LIVES<=0 THEN GOTO 9500
9025 CLS
9030 GOTO 210
9500 IF HS<S THEN LET HS=S
9510 PRINT AT 10,0;"HIT ANY KEY
TO RESTART"
9520 IF INKEY$="" THEN GOTO 9520
9530 LET S=0
9540 CLS
9550 LET T=180
9560 LET LIVES=3
9570 GOTO 8
9998 LET A$="MUNCHER II"
9999 SAVE A$
```

THE RIGHT CHORD

Having difficulty getting your fingers round those tricky guitar chords? Bring you ZX81 into play.



```
SAMPLE SCREEN.
--------------

GUITAR CHORD FINDER.

 E A D G B E      C  = 1
                  D  = 2
                  E  = 3
                  F  = 4
                  G  = 5
                  A  = 6
                  DM  = 7
                  EM  = 8
                  AM  = 9
                  G7  = 10

                  C

                INPUT CHORD
                 NUMBER.
```

When RUN, this program draws a simulation of a guitar neck and asks the user to input a number corresponding to the particular chord you wish to see on the screen.

GETTING IN TUNE...

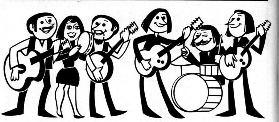
The program asks the user to input a number from one to ten and then the chord corresponding to the number will be displayed with the relevant finger positions. The chord name is displayed in a box beside the diagram and the program then waits for a further input.

You could add two more chords to the published program by changing line 82 to read:

82 IF C<1 OR C>12 THEN GOTO 80

and enter the co-ordinates for the new chords in the subroutines at lines 5500 and 6000. Then, you would have to alter the menu in line 7065. The chords included in the program as it stands are C, D, E, F, G, A, Dm, Em, Am, and G7.

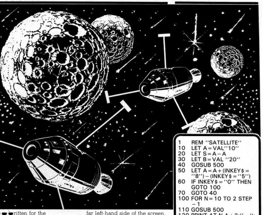
Now if only someone could write a program to tune the guitar... ▶



```
  10 REM GUITAR CHORD FINDER
  15 REM BY C.A.READ.
  20 LET X=15
  30 LET Y=17
  40 GOSUB 7000
  80 INPUT C
  82 IF C<1 OR C>10 THEN GOTO 80
  85 GOSUB 7000
  90 GOSUB C*500
 100 PRINT AT A,B;"O"
 110 PRINT AT C,D;"O"
 120 PRINT AT E,F;"O"
 130 GOTO 80
 490 REM C CHORD
 500 LET A=4
 510 LET B=10
 520 LET C=7
 530 LET D=6
 540 LET E=10
 550 LET F=4
 560 PRINT AT X,Y;"C"
 570 RETURN
 990 REM D CHORD
1000 LET A=7
1010 LET B=8
1020 LET C=10
1030 LET D=10
1040 LET E=7
1050 LET F=12
1060 PRINT AT X,Y;"D"
1070 RETURN
1500 REM E CHORD
1510 LET A=7
1520 LET B=4
1530 LET C=7
1540 LET D=6
1550 LET E=4
1560 LET F=8
1570 PRINT AT X,Y;"E"
1580 RETURN
2000 REM F CHORD
2010 LET A=10
2020 LET B=6
2030 LET C=7
2040 LET D=8
2050 LET E=4
2060 LET F=10
2070 PRINT AT 4,12;"O"
2080 PRINT AT X,Y;"F"
2090 RETURN
2500 REM G CHORD
2510 LET A=10
2520 LET B=2
2530 LET C=7
2540 LET D=4
2550 LET E=10
2560 LET F=12
2570 PRINT AT X,Y;"G"
2580 RETURN
3000 REM A CHORD
3010 LET A=7
3020 LET B=6
3030 LET C=7
3040 LET D=8
3050 LET E=7
3060 LET F=10
3070 PRINT AT X,Y;"A"
3080 RETURN
3500 REM DM CHORD
3510 LET A=7
3520 LET B=8
3530 LET C=10
3540 LET D=10
3550 LET E=4
3560 LET F=12
3570 PRINT AT X,Y;"DM"
3580 RETURN
4000 REM EM CHORD
4010 LET A=7
4020 LET B=4
4030 LET C=7
4040 LET D=6
4050 LET E=7
4060 LET F=6
4070 PRINT AT X,Y;"EM"
4080 RETURN
4500 REM AM CHORD
4510 LET A=7
4520 LET B=6
4530 LET C=7
4540 LET D=8
4550 LET E=4
4560 LET F=10
4570 PRINT AT X,Y;"AM"
4580 RETURN
5000 REM G7 CHORD
5010 LET A=10
5020 LET B=2
5030 LET C=7
5040 LET D=4
5050 LET E=4
5060 LET F=12
5070 PRINT AT X,Y;"G7"
5080 RETURN
7000 PRINT AT 0,0;"GUITAR CHORD
FINDER."
7010 PRINT "▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
7015 PRINT "  E A D G B E"
7020 FOR T=1 TO 6
7030 PRINT "  █▀█▀█▀█▀█▀█"
7040 PRINT "  █ █ █ █ █ █"
7050 PRINT "  █ █ █ █ █ █"
7060 NEXT T
7065 PRINT AT 2,16;"C = 1",,"D =
 2",,"E = 3",,"F = 4",,"G = 5",,
"A = 6",,"DM = 7",,"EM = 8",,"AM
 = 9",,"G7 = 10"
7070 PRINT AT 19,15;"INPUT CHORD
",,"NUMBER."
7080 PRINT AT 14,16;"█▀▀▀█",,"█
█",,"█▄▄▄█"
7090 RETURN
```

ENEMY SATELLITE

Don't let the enemy get your secrets.



Written for the unexpanded ZX81, the object of the game is to shoot the satellites out of the sky before they can relay information on your military bases back to the enemy.

As the enemy satellites traverse across the top of the screen, it is your job to move the ground to air missile launcher until you can get a good shot at the enemy device. Movement of the missile launcher is achieved by pressing the '5' key to go left and the '8' key to move right. Key '0' is used to launch a missile at the satellite.

Should a satellite reach the far left-hand side of the screen, it is assumed that the device has excaped destruction and has thus relayed all of the secret data on your military establishments. This is achieved in line 520 with the 'Q', and an error 2 message denotes the end of the game.

To hit a satellite, you must aim for the '0' character of the device. if you do manage to destroy the satellite, line 160 causes an explosion to be displayed: this is followed by a PAUSE statement causing the screen to blink (just to add a touch of realism). When the game finishes, your score is displayed.

```
1   REM ''SATELLITE''
10  LET A=VAL''10''
20  LET S=A-A
30  LET B=VAL ''20''
40  GOSUB 500
50  LET A=A+(INKEY$=
    ''8'')-(INKEY$=''5'')
60  IF INKEY$=''0'' THEN
    GOTO 100
70  GOTO 40
100 FOR N=10 TO 2 STEP
    -1
110 GOSUB 500
120 PRINT AT N,A+2;''*'';
    AT N+1,A+2;'' ''
130 NEXT N
140 IF A=INT B THEN
    GOTO 160
150 GOTO 40
160 PRINT AT 2,B;''>)X (<''
170 PAUSE 50
180 LET S=S+B
190 CLS
200 GOTO 30
500 LET B=B-0.25
510 PRINT AT 2,INT B;''
    =0= '';AT 11,A;
    ''▄ ▀*▀ ▄''
520 IF B<0.2 THEN PRINT
    AT 2,0;''GAME OVER
    - '';INT S;Q
530 RETURN
```

Timothy Parnell

ON TARGET

Two great programs – Target and Missile Launcher.

When RUN, Target displays a target and a randomly placed star near the centre of the screen. You must then, using the 'N' key to move left and the 'M' key to move right, position the target so that the star is in the middle of it. Pressing the 'X' key will then fire at the star.

Confirmation of a hit on the star is given by the star turning to inverse video. The program ends if you fail to hit a star.

You set the difficulty of the game yourself at the beginning of the game; if you input a value below 15, the game is very hard indeed! The difficulty is calculated in the FOR. . . TO loop in line 50.

MISSILE LAUNCHER

In this program, you control a ground-based missile launcher with which you can Fire missiles at oncoming alien attackers.

You can move your missile launcher using the 'N' key to go right and the 'M' key to go left. The 'Z' key is used to launch the missile. You can cheat by holding the 'Z' key down all the time as you move, but this will show up at the end of the game when the number of shots you took is displayed.

The game can be altered so that the missile rises at a faster rate. This is done by changing line 150 to read:

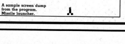
```
150 IF B<= 19 THEN LET
     B=B-2
```

If you make this change, you should also alter line 70 to read:

```
70 IF B<=0 THEN LET
    B=20
```

The listing for Missile launcher.

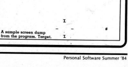
```
   5 LET S=0
  10 LET X=15
  20 LET Z=0
  30 LET Y=0
  40 LET B=20
  50 PRINT AT B,X;"▌"
  60 PRINT AT B+1,X;"▞ "
  70 IF B=0 THEN LET B=20
  80 PRINT AT Y,Z-1;"▄█▖"
  90 LET Z=Z+1
 100 IF Z=31 THEN LET Y=Y+5
 110 IF Z=31 THEN LET Z=0
 120 IF INKEY$="N" THEN LET X=X-
1
 130 IF INKEY$="M" THEN LET X=X+
1
 140 IF INKEY$="Z" THEN LET B=19
 145 IF INKEY$="Z" THEN LET S=S+
1
 150 IF B<=19 THEN LET B=B-1
 160 IF Y=20 AND Z=X THEN GOTO 2
90
 170 IF X=Z AND B+1=Y THEN GOTO
300
 180 CLS
 190 GOTO 50
 290 PRINT AT 10,6;"ALIENS HAVE
LANDED"
 298 STOP
 300 PRINT AT 10,0;"  YOU HIT TH
E ALIEN IN ";S;" SHOTS"
```



A sample screen dump from the program. Missile launcher.

The listing for Target.

```
   5 PRINT "INPUT DIFFICULTY"
   6 INPUT D
   7 CLS
  10 LET S=0
  20 LET X=10
  30 LET Y=10
  40 LET V=INT (RND*28)
  50 FOR T=1 TO D
  60 PRINT AT X,Y;" I ";AT X+4,Y
;" I ";AT X+2,Y+2;" - ";AT X+2,Y
-2;" - "
 100 LET Y=Y+(INKEY$="M")-(INKEY
$="N")
 120 IF INKEY$="X" AND Y+2=V+1 T
HEN GOTO 160
 130 PRINT AT 12,V;"*"
 140 NEXT T
 150 PRINT AT 0,10;"YOU HIT ";S
 155 STOP
 160 PRINT AT 12,V;"■"
 170 LET S=S+1
 180 PAUSE 50
 185 CLS
 190 GOTO 20
```



```
YOU HIT 3
```

A sample screen dump from the program. Target.

"Run more than ten tasks on a ZX81-FORTH ROM?"

Sure! More than 10 tasks simultaneously and, in some cases, up to 300 times faster! That's what replacing the basic ROM with the new FORTH does for the ZX81 – and more!

The brains behind the breakthrough belong to David Husband, and he's building Skywave Software on the strength of it. Already orders are flooding in and it's easy to see why.

The ZX81-FORTH ROM gives you a totally new system. In addition to multi-tasking and split screen window capability, you can also edit a program while three or four others are executing, schedule tasks to run from 50 times a second to once a year, and with a further modification switch between FORTH and BASIC whenever you like.

The ZX81-FORTH ROM gives you a normal keyboard with a 64 character buffer and repeat, it supports the 16k, 32k, 64k RAM packs, it is fig-FORTH compatible and it supports the ZX printer.

The price, too, is almost unbelievable. As a ''fit it yourself Eprom'', complete with manual, it's just £25 + VAT.
Add £2 p&p UK (£5 Europe, £10 outside Europe) and send your order to the address below.

Skywave
SOFTWARE
David Husband
73 Curzon Road, Bournemouth,
BH1 4PW, ENGLAND.
Tel: (0202) 302385.
International +44 202 302385.

~~**50 games for the Sinclair Spectrum, 16/48K. There are fifty super high quality games on one tape!!**~~

STOP PRESS:

There are now sixty games on this fantastic compilation, yes 60 super high resolution arcade games that you will have hours and hours of enjoyment from, guaranteed!!!

60 of the best games around including: Pacman, Invaders, Frogger, Fruit Machine, Defender, Lunar Lander, Pontoon, plus much more!! All on one tape for only £9.95.

50 games for the ZX81 — £6.99. An amazing simulation of the sidewinder arcade fruit machine, features include random holds and nudges, flip-winding reels to a winning line. Visually stunning graphics to 'state of the art' machine code, true arcade sound effects.
Introductory price £5.50.

ONLY AVAILABLE BY MAIL ORDER FROM:

ANCO SOFTWARE (DEPT ASP)
25 Corsewall Street, Coatbridge,
Scotland ML5 1PX

ANCO SOFTWARE . . . THE ULTIMATE EXPERIENCE

It's easy to complain about an advertisement. Once you know how.

One of the ways we keep a check on the advertising that appears in the press, on posters and in the cinema is by responding to consumers' complaints.

Any complaint sent to us is considered carefully and, if there's a case to answer, a full investigation is made.

If you think you've got good reason to complain about an advertisement, send off for a copy of our free leaflet.

It will tell you all you need to know to help us process your complaint as quickly as possible.

The Advertising Standards Authority.
If an advertisement is wrong, we're here to put it right.
ASA Ltd, Dept 1 Brook House,
Torrington Place, London WC1E 7HN

This space is donated in the interests of high standards of advertising.

ELECTRIC

SOFTWARE

COMES IN SUPERB PACKAGING

Software at prices people can afford

48K SPECTRUM

THE PRICE REVOLUTION IS HERE!

AT LAST AN ADVENTURE THAT'S FUN TO PLAY

PRESENTS

THEY SAY THE WORLD WILL DIE IN FIRE AND ICE

A massive 48K graphic adventure with a sense of humour for the Spectrum.

IMMEDIATE DELIVERY SERVICE
EG. SEND US ORDER MONDAY
WE RECIEVE MONEY TUESDAY
YOU RECIEVE PROGRAM WEDNESDAY!

SEND YOUR ORDERS TO

ELECTRIC SOFTWARE
13 CLARENDON RD
LONDON E.11 1B2
TEL. 01-247 0158
DEALER ENQUIRIES WELCOME

100% MACHINE CODE

Special features include:
Abbreviations and long sentences both acceptable.
Immediate response time. Graphic locations.
An amazing sense of humour.
Full instructions on loading

*Special Bonus 30 minutes of tape for you to record your own programmes

SPECIAL INTRODUCTORY PRICE – JUST 2-99!!!

Norman Brooks

SMAUG'S LAIR

Can you destroy the dragon's lair?

You are the Gunner Captain on a mission to destroy the dragon's castle.

You have five cannon and 12 cannonballs at your disposal . . . but . . . beware . . . the sound of your guns will wake the dragon. When he flies, time stands still and although he has poor eyesight, he will breathe fire and try to destroy all of your weapons. If he does . . . ***run for your life!!!***

To dislodge the most bricks, hit the castle as high as you can by carefully choosing how much gunpowder to use and what angle to set the cannon at. If 6,000 bricks fall then the castle is in ruins and Smaug must seek a new wormhole. Good luck!!!

CANNON AND BALL

When you first RUN this program you are asked if you require instructions in the usual way. If you reply 'yes' (Y) you are given a brief introduction to the program to get the adrenalin up.

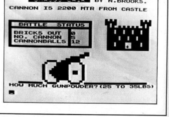
Then, the 'Battle Status' screen is shown and is returned to throughout the game. You are shown the current scores, and the castle will be demolished in proportion to the number of bricks you have knocked out. At the top of the screen, you see how far the computer has set the cannon from the castle ready for your first shot (this is random between 500 and 2,300 metres). You are then invited to enter the size of the gunpowder charge (25 to 35lbs) and what angle of elevation you wish the cannon to bet set to (0 to 89 degrees).

Should the cannonball reach its target (the castle is 400 metres high) then the Y co-ordinate of the cannonball plot determines how many bricks are knocked out. (See program lines 1071 to 1079.) When in flight, the dragon may obscure the ball but it will always re-appear in the same position after the dragon has passed — unless, of course, it burns the cannonball to a cinder! You are finally asked if you wish to continue the game. Answering 'no' (N), puts you in retreat and the dragon wins.

At the end of the game (either with 6,000 bricks out of the castle or no cannon/cannonballs left) there is an appropriate 'picture show' to watch which I leave you to discover.

The odds slightly favour the Gunner Captain once an appreciation of charge and angle has been gained. Bear in mind that an angle of 45 degrees will send the ball the maximum distance for any choice of charge (set at which, the castle is seldom damaged incidentally) and an angle of say 60 degrees will result in the same final range as that of 30 degrees, ie each 15 degrees either side of 45 degrees. However, the castle presents a smaller target when dropped on from above rather than hit broadside on, so for a more difficult game try always chosing angles over 45 degrees. I have yet to win doing this!

The more gunpowder you use, the faster the cannonball leaves the cannon and hence the further the cannonball will fly. Similarly, the angle of elevation affects the range. (I have used Newton's formulae for the motion of projectiles from my old 'A' level maths notes for this part of the program and have consulted books on cannons for details of range and charge to make the scientific aspects of the program as accurate as possible).



```
SMAUGS LAIR BY N.BROOKS.
CANNON IS 2200 MTR FROM CASTLE

BATTLE   STATUS
BRICKS OUT    0
NO. CANNON    5
CANNONBALLS   12

HOW MUCH GUNPOWDER?(25 TO 35LBS)
```

A graphical display of the cannon being adjusted to the angle chosen is shown next. (I make no apology for the slowness of this part of the program as I wanted to introduce the concept of 'angles' to my young family. Adding lines 2205 FAST and 2910 SLOW will alleviate any boredom.)

INTO BATTLE

The main action comes on the 'Battlefield' screen which follows on from the previous screen automatically. The cannon is displayed at the computer-selected distance from the castle and the cannonball is fired. Every fifth plot of the cannonball, the dragon flies from the castle. Unbeknown to the player, it has chosen at random whether to attack the cannon or the cannonball. If the dragon chooses the cannon there is a 50% chance of a hit. If the cannonball is selected, then the odds of a hit are only 20%.

VARIABLES ON A THEME

Listed here are the main variables used throughout the game:

BRIC	The number of bricks knocked out of the castle (6,000 bricks knocked out will win the game for you).
BALL	The number of cannonballs remaining (zero cannonballs left will result in you losing the game).
GUNS	The number of cannon you have remaining (zero cannon left will result in you losing the game).
ROUT	The 'I give up' code.
M	The distance between the cannon and the castle.
N	The random number used to generate the variable, M.
G	If G is equal to one, the dragon attacks the cannonball only.
X and Y	The co-ordinates of the cannonball.
A	The input for the angle of the cannon in radians.
B	The angle converted into radians.
U	The input for the amount (in lbs) of gunpowder used.

Other variables are used but they only exist to aid the running of the program.

```
  10 REM "SMAUGS LAIR"
  12 PRINT TAB 4;"SMAUGS LAIR
BY N.BROOKS."
  14 PRINT
  16 PRINT "DO YOU WANT INSTRUCT
IONS? (Y/N)"
  18 IF INKEY$="" THEN GOTO 18
  20 LET E$=INKEY$
  22 IF E$="Y" THEN GOSUB 9000
  50 LET BRIC=0
  60 LET GUNS=5
  70 LET BALL=12
  75 LET ROUT=0
  80 CLS
  90 RAND
 150 LET N=INT (RND*19)
 200 LET M=N*100+500
 210 LET G=0
 225 GOSUB 8000
 340 INPUT A
 350 IF A>89 THEN GOTO 8950
 400 GOSUB 2000
 440 REM **MAIN PROGRAM**
 450 CLS
 480 PRINT TAB 4;"SMAUGS LAIR
BY N.BROOKS."
 500 LET B=A*PI/180
 650 PRINT AT 15,27;" █ "
 660 PRINT AT 16,27;" █ "
 665 PRINT AT 17,27;" █ "
 670 PRINT AT 18,27;" █ "
 680 PRINT AT 19,27;" █ "
 690 PRINT AT 20,2;"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀
▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀ ▀▀▀ ▀▀"
 700 PRINT AT 19,20-N;"O▛"
 710 GOSUB 6000
 720 LET X1=0
 730 LET Y1=0
 750 FOR X=100 TO M+400 STEP 100
 850 LET Y=(X*TAN B)-(X**2/(U*20
0*COS B*COS B))
 860 IF Y<=0 AND X<M THEN GOTO 1
100
 865 IF Y<=0 AND X>=M+150 THEN G
OTO 1050
 870 PLOT (X/50)+43-2*N,(Y/50)+5
 890 IF X1>0 THEN UNPLOT (X1/50)
+43-(2*N),(Y1/50)+5
 910 IF X/250=INT (X/250) AND X<
>0 THEN GOSUB 3000
 920 LET X1=X
 930 LET Y1=Y
1000 IF X>=M AND X<(M+150) AND Y
<400 THEN GOTO 1070
1020 NEXT X
1050 PRINT AT 12,5;"*CANNONBALL
OVERSHOT*"
1060 GOTO 1120
1070 PRINT AT 16,25;"**HIT**";AT
 15,25;"*******";AT 17,25;"*****
**"
1071 IF Y<=400 AND Y>350 THEN LE
T BRIC=BRIC+1500
1072 IF Y<=350 AND Y>300 THEN LE
T BRIC=BRIC+1000
1074 IF Y<=300 AND Y>200 THEN LE
T BRIC=BRIC+750
1076 IF Y<=200 AND Y>100 THEN LE
T BRIC=BRIC+500
1078 IF Y<=100 THEN LET BRIC=BRI
C+250
1079 PRINT AT 12,2;"*BALL STRUCK
 ";INT (Y);" METRES UP*"
1080 GOTO 1120
1100 PRINT AT 12,6;"*CANNONBALL
SHORT*"
1120 LET BALL=BALL-1
1125 PRINT AT 21,0;"DO YOU WANT
TO CONTINUE?(""N""=NO)"
1127 IF INKEY$="" THEN GOTO 1127
1135 IF INKEY$="N" THEN LET ROUT
=1
1145 GOTO 80
2000 REM **ADJUST CANNON**
2020 CLS
2050 PRINT "GUNNER SERGEANT...LO
AD CHARGE   OF ";U;" LBS AND SET
 CANNON TO      ";A;" DEGREES"
2070 GOSUB 2992
2080 PRINT
2100 PRINT "YES......SIR"
```

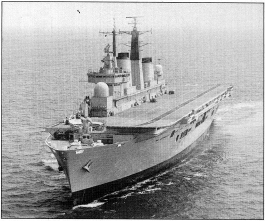
```
2150 PRINT AT 16,6;" ▄▀▀▄ ████████
█████████"
2160 PRINT AT 17,6;"█▖ ▗█████████
"
2165 PRINT AT 17,13;6-GUNS
2170 PRINT AT 18,6;"▌ ▀▄▀ ████████

2180 PRINT AT 19,6;"▄▀▀▄▄███"
2190 PRINT AT 20,6;"▀▄▄▀▀▄██"
2200 PRINT AT 21,2;"▄▄▄▄▀▀▀▀▄▄████
███████████████████"
2202 GOSUB 2992
2210 GOSUB 5000
2220 IF A>45 THEN GOTO 2650
2270 FOR X=1 TO 20
2280 LET Y=X*TAN (A/180*PI)
2320 UNPLOT (X+35),11
2330 UNPLOT (X+35),10
2350 PLOT (X+35),(Y+11)
2360 PLOT (X+35),(Y+10)
2550 NEXT X
2500 GOTO 2910
2650 LET A=90-A
2700 FOR Y=1 TO 20
2720 LET X=Y*TAN (A/180*PI)
2740 UNPLOT (Y+35),11
2760 UNPLOT (Y+35),10
2780 PLOT X+35,Y+11
2800 PLOT X+36,Y+11
2850 NEXT Y
2900 LET A=90-A
2920 GOSUB 2992
2930 PRINT AT 6,0;"CANNON SET AT
"
2940 PRINT A;" DEGREES...SIR"
2945 PRINT
2950 GOSUB 2992
2965 PRINT "*********"
2970 PRINT "* FIRE *"
2975 PRINT "*********"
2980 GOSUB 2992
2990 RETURN
2992 FOR R=1 TO 25
2994 NEXT R
2996 RETURN
3000 REM **FLY DRAGON**
3020 LET O=INT (RND*2)
3030 IF G=1 THEN LET O=1
3050 IF O=0 THEN LET P=N+4+INT (
RND*4)
3060 IF O=1 THEN LET P=25-INT ((
(X/50)+43-(2*N))/2)+INT (RND*5)
3070 IF P<0 THEN LET P=INT (RND*
5)
3080 IF P>24 THEN LET P=24-INT (
RND*4)
3100 FOR R=1 TO 5
3110 PRINT AT 15-R,26;"▀▄▀"
3120 PRINT AT 15-R,26;"   "
3130 PLOT (X/50)+43-2*N,(Y/50)+5
3140 NEXT R
3150 FOR R=1 TO 4
3160 PRINT AT 10-R,26;"▄█"
3170 PRINT AT 9-R,25;"▀█▀█▀"
3180 PRINT AT 10-R,26;"  "
3190 PRINT AT 9-R,25;"     "
3200 PLOT (X/50)+43-2*N,(Y/50)+5
3500 NEXT R
4000 FOR Q=26 TO 26-P STEP -1
4020 PRINT AT 7,Q-1;"▀▒▀"
4030 PRINT AT 7,Q+2;" "
4040 PRINT AT 6,Q-1;"▀█▀"
4060 PRINT AT 6,Q+2;" "
4070 PRINT AT 5,Q-2;"▀████▀"
4080 PRINT AT 5,Q+3;" "
4090 PRINT AT 4,Q-2;"▄▀▄▀▄▀▄"
4120 PRINT AT 4,Q+3;" "
4140 PRINT AT 3,Q-2;"▗▌▐ ▌▖"
4160 PRINT AT 3,Q+3;" "
4180 PRINT AT 2,Q-2;"▌   ▐"
4200 PRINT AT 2,Q+3;" "
4240 PRINT AT 1,Q-2;"▖   ▗"
4250 PRINT AT 1,Q+3;" "
4280 PLOT (X/50)+43-2*N,(Y/50)+5
4300 NEXT Q
4350 FOR R=7 TO 19
4400 PRINT AT R,Q+1;"▒"
4450 PRINT AT R,Q+1;" "
4460 NEXT R
4500 IF G=0 AND (Q+1=21-N OR Q+1
=20-N) THEN GOSUB 6600
4600 IF Q+1=INT (((X/50)+43-(2*N
))/2) THEN GOTO 6700
4760 FOR R=1 TO 7
4800 PRINT AT R,Q-1;"     "
4810 PLOT (X/50)+43-2*N,(Y/50)+5
4820 NEXT R
4990 RETURN
5000 REM **ANGLE OF TRAGECTORY**
5050 FOR R=0 TO 2
5100 PRINT AT 16-R*3,28;R*15
5200 NEXT R
5250 PRINT AT 6,28;45
5300 FOR R=0 TO 2
5350 PRINT AT 5,16+(7-R*3);60+R*
15
5400 NEXT R
5420 PRINT AT 3,19;"ELEVATION"
5440 LET R$="IN  DEGREES"
5460 FOR R=1 TO 11
5470 PRINT AT 3+R,31;R$(R)
5480 NEXT R
5500 RETURN
6000 REM **FIRE**
6050 LET A$=" BOOM"
6060 LET B$=" "
6100 FOR R=1 TO 5
6150 PRINT AT 19,21-N+R;A$(R)
6200 NEXT R
6250 FOR R=1 TO 5
6300 PRINT AT 19,21-N+R;B$
6350 NEXT R
6500 RETURN
6600 REM *CANNON DESTROYED*
6610 LET GUNS=GUNS-1
6620 PRINT AT 21,0;"*SMAUG HAS D
ESTROYED THE CANNON*"
6630 LET G=1
6650 RETURN
6700 LET BALL=BALL-1
6710 PRINT AT 12,0;"*SMAUG HAS B
URNT THE CANNONBALL*"
6720 GOTO 1125
7000 REM *WIN/LOSE ROUTINES*
7100 PRINT AT 14,19;"CAPTAIN WIN
S"
7110 PRINT AT 15,19;"THE CASTLE
IS"
7120 PRINT AT 16,19;"DESTROYED A
ND"
7130 PRINT AT 17,20;"SMAUG MUST"
7140 PRINT AT 18,20;"SEEK A NEW"
7150 PRINT AT 19,21;"WORMHOLE."
7200 FOR R=11 TO 2 STEP -1
7210 PRINT AT R,24;"▀▄▀"
7220 PRINT AT R,24;"   "
7230 NEXT R
7240 FOR R=24 TO 2 STEP -1
7250 PRINT AT 2,R;"▀▄▀"
7260 PRINT AT 2,R;"   "
7270 NEXT R
7280 FOR R=3 TO 31
7290 IF R<=16 THEN PRINT AT 2,R;
"▖"
7295 IF R>16 THEN PRINT AT 2,R;"
"
7300 PRINT AT 2,R;"  "
7305 NEXT R
7340 STOP
7350 PRINT AT 13,20;"SMAUG WINS"
7370 IF GUNS=0 THEN GOTO 7410
7380 IF ROUT=1 THEN GOTO 7400
7390 PRINT AT 14,18;"YOU ARE OUT
 OF"
7395 PRINT AT 15,19;"CANNONBALLS
."
7398 GOTO 7440
7400 PRINT AT 14,18;"YOU ARE BEA
TEN"
7405 PRINT AT 15,18;"AND IN RETR
EAT"
7408 GOTO 7440
7410 PRINT AT 14,18;"ALL THE CAN
NON"
7420 PRINT AT 15,18;"ARE DESTROY
ED."
7440 PRINT AT 17,18;"************
***"
7450 PRINT AT 18,18;"*  FLEE FOR
  *"
7460 PRINT AT 19,18;"* YOUR LIVE
S *"
7470 PRINT AT 20,18;"************
***"
```

```
7472 PRINT AT 11,24;"▚▞"
7473 PRINT AT 11,24;"  "
7474 FOR R=24 TO 17 STEP -1
7475 PRINT AT 12,R;"▚▞"
7476 PRINT AT 12,R;"  "
7477 NEXT R
7478 PRINT AT 11,24;"█"
7479 PRINT AT 12,21;"▀▀▀▀▀"
7480 FOR R=1 TO 8
7482 PRINT AT 12+R,17-2*R;"▚▞"
7484 PRINT AT 12+R,17-2*R;"  "
7486 NEXT R
7487 DIM R$(7,5)

7488 LET R$(1)="▘   ▝"
7489 LET R$(2)="▌   ▐"
7490 LET R$(3)="▌▐ ▌▐"
7491 LET R$(4)="▙███▟"
7492 LET R$(5)=" ███ "
7493 LET R$(6)=" ▝█▘ "
7494 LET R$(7)="  ▒  "
7495 FOR R=1 TO 7
7496 PRINT AT 21-R,1;R$(8-R)
7497 NEXT R
7500 DIM D$(13,10)
7510 LET D$(1)="▜      ▛"
7515 LET D$(2)=" ▌    ▐ "
7520 LET D$(3)=" ▌    ▐ "
7525 LET D$(4)=" ▌ ▐▌ ▐ "
7530 LET D$(5)=" ▌ ▐▌ ▐ "
7535 LET D$(6)=" ▙▄██▄▟ "
7540 LET D$(7)="  █▘▝█  "
7545 LET D$(8)="  █████ "
7548 LET D$(9)="   ███  "
7550 LET D$(10)="    ███   "
7560 LET D$(11)="     █    "
7570 LET D$(12)="    ▀▒▀   "
7580 LET D$(13)="     ▒    "
7650 FOR R=1 TO 13
7560 PRINT AT 21-R,1;D$(14-R)
7670 NEXT R
7700 DIM A$(20,18)
7705 LET A$(1)="▚  ██   ██   
▞ "
7708 LET A$(2)=" ▚  ██   ██  ▟
▌ "
7710 LET A$(3)="  █ ██   ██  █
 "
7712 LET A$(4)=" █  ██   ██  ▐
▌ "
7715 LET A$(5)=" █ ████████████▐
▌ "
7716 LET A$(6)="█ ████████████
█ "
7720 LET A$(7)="█ ██▄████▄██
█ "
7722 LET A$(8)=" █ ██▀████▀██▐
▌ "
7725 LET A$(9)=" ███████████████
▘ "
7726 LET A$(10)="  █████████████
▚ "
7730 LET A$(11)="   ████████████
 "
7731 LET A$(12)="  ▄█████████████
▚ "
7732 LET A$(13)=" ▟   ██ ██ ██
▚ "
7735 LET A$(14)=" ▚   ████████
▞ "
7738 LET A$(15)="  ▚    █████
▘ "
7740 LET A$(16)=" ▟▙    █████   ▖
▘ "
7742 LET A$(17)="▘▟▐  ████████  ▌
▝▖ "
7745 LET A$(18)=" ▖▖  ████████  ▖
▖ "
7748 LET A$(19)="        ████
 "
7750 LET A$(20)="        ████
 "
7760 FOR R=1 TO 20
7770 PRINT AT 22-R,0;A$(21-R)
7775 NEXT R
7990 STOP
8000 REM **DISPLAY SCENE**
8010 PRINT TAB 4;"SMAUGS LAIR
BY N.BROOKS."
8050 PRINT AT 5,0;"▀▀▀▀▀▀▀▀▀▀▀▀▀
▀▀▀▀ ▛▀▜      ▛▀▜"
8055 PRINT "▌ BATTLE  STATUS ▌
▜▛.....▜▛"
8060 PRINT "▛▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▜▀▀▀▀▜
▀▜▛▀▜▛▀▜▛▀▀"
8065 PRINT "▌BRICKS OUT   ▌    ▌
▚▚▚▚▚▚▚▚▚"
8070 PRINT "▌NO. CANNON   ▌    ▌
▒▒ ▒ ▒ ▒▒▒"
8075 PRINT "▌CANNONBALLS  ▌    ▌
▀▜▛▀▀▀▜▛▀▀"
8080 PRINT "▙▄▄▄▄▄▄▄▄▄▄▄▄▄▄▙▄▄▄▄▟
▀▀▀▀+▀▀▀▀"
8085 PRINT AT 12,21;"▀▀▀▀▀▀▀▀▀▀"
8100 PRINT AT 13,15;"▄█▄"
8105 PRINT AT 14,12;"▀▀▀█▀▀▀"
8110 PRINT AT 15,7;"  ▄▄█▀▀▀▀▄   "
8115 PRINT AT 16,7;"▐ ▟███▌  ▐ "
8120 PRINT AT 17,7;"▐▟████▌ █ ▌"
8125 PRINT AT 18,7;"▐▜████▌ ▀ ▌"
8130 PRINT AT 19,7;"▐ ▜███▙ ▄▟ "
8135 PRINT AT 20,2;"     ▀▀▀▀▀▀▀▀  ▚▄
▘▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
8138 IF BRIC<250 THEN GOTO 8155
8140 LET E$="        "
8142 LET E=INT (BRIC/1000)
8144 LET F=INT ((BRIC-E*1000)/20
0)
8145 FOR R=0 TO E
8146 PRINT AT 5+R,23;E$
8147 NEXT R
8148 FOR R=0 TO F
8149 PRINT AT 6+E,23+R;E$(1)
8150 NEXT R
8152 PRINT AT 12,23;"▀▀▀▀▀▀"
8155 PRINT AT 8,13;BRIC
8160 PRINT AT 9,13;GUNS
8170 PRINT AT 10,13;BALL
8200 IF BRIC>=6000 THEN GOTO 710
0
8300 IF GUNS=0 THEN GOTO 7350
8400 IF BALL=0 THEN GOTO 7350
8500 IF ROUT=1 THEN GOTO 7350
8900 PRINT AT 2,0;"CANNON IS ";M
;" MTR FROM CASTLE"
8910 PRINT AT 21,0;"HOW MUCH GUN
POWDER?(25 TO 35LBS)"
8912 INPUT U
8915 IF U<25 OR U>35 THEN GOTO 8
910
8950 PRINT AT 21,0;"ENTER CANNON
 ANGLE (MAX 89 DEG)"
8990 RETURN
9000 REM **INSTRUCTIONS**
9020 PRINT AT 2,0;"YOU ARE THE G
UNNER CAPTAIN ON A"
9025 PRINT "MISSION TO DESTROY T
HE DRAGONS"
9030 PRINT "CASTLE."
9040 PRINT "YOU HAVE 5 CANNON AN
D 12 CANNON"
9045 PRINT "BALLS AT YOUR DISPOS
AL...BUT..."
9050 PRINT "BEWARE...THE SOUND O
F YOUR GUNS "
9055 PRINT "WILL WAKE THE DRAGON
"
9065 PRINT "WHEN HE FLIES TIME S
TANDS STILL"
9070 PRINT "AND ALTHOUGH OF POOR
 EYESIGHT, "
9075 PRINT "HE WILL BREATH FIRE
AND TRY TO"
9080 PRINT "DESTROY ALL OF YOUR
WEAPONS. IF HE DOES...RUN FOR YO
UR LIFE."
9090 PRINT "TO DISLODGE MOST BRI
CKS,HIT THE"
9095 PRINT "CASTLE AS HIGH AS YO
U CAN BY"
9096 PRINT "CAREFULLY CHOOSING H
OW MUCH GUN-"
9098 PRINT "POWDER TO USE AND WH
AT ANGLE TO"
9099 PRINT "SET THE CANNON AT.IF
 6000 BRICKS"
9100 PRINT "FALL THEN THE CASTLE
 IS IN RUINS"
9110 PRINT "AND SMAUG MUST SEEK
A NEW HOLE."
9970 PRINT "*GOOD LUCK* (ANY KEY
 TO START)"
9980 PAUSE 40000
9990 RETURN
```

M J Downie

NAVAL MANOEUVRES

Can you stay afloat in this ZX81 version of the popular board game?



Written for the 16K ZX81, this version of 'Battleships' involves you taking on the computer in a naval conflict.

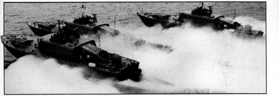
IN THE SWIM

As in the game we all know and love, the idea of the game is to sink your opponent's fleet before it manages to sink yours.

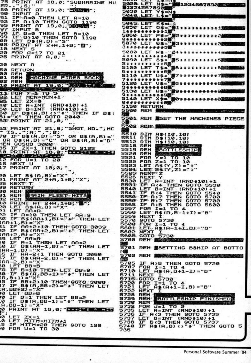
When you first run the program, the computer first provides you with a few brief instructions on how to play the game. When you are sure you understand the workings of the game, the computer sets up its fleet on the right-hand playing area. Once this has been done, you position your own fleet.

You do this under instruction from the computer which invites you to give the co-ordinates of each ship in your fleet. You must position one battleship (represented by four squares), two cruisers (three squares), three destroyers (two squares) and four submarines (one square each). You position your fleet by first inputting the row co-ordinate on your 10 by 10 playing grid on the left-hand part of the screen.

You begin the game by having three shots at the computer's fleet. This is done in a similar manner to how you inputted your fleet, in that the computer asks you first for the row co-ordinate followed by the column co-ordinate. The computer will then have three shots at your fleet and so on. All shots and hits are shown clearly on the screen, and the winner is the one who destroys the other's fleet first. The player has the same chance of winning as the computer, so have fun!



```
   1 REM
   2 REM  COMPILED BY M.J AND
   3 REM  J DOUNIE, 28 DEC 1982
   4 REM
   6 CLS
   8 CLEAR
   9 GOSUB 7500
  10 FAST
  15 GOSUB 5000
  18 GOSUB 5500
  30 GOSUB 7000
  40 GOSUB 1000
  45 LET HITP=0
  47 LET HITM=0
  50 LET PLR=0
  51 LET MCN=0
  55 GOSUB 500
  56 FOR U=18 TO 21
  57 PRINT AT U,0;"
                                ";
  58 FOR I=1 TO 5
  59 NEXT I
  60 PRINT AT U,0;"
                                ";
  61 NEXT U
  70 FOR U=1 TO 100
  75 NEXT U
  90 GOSUB 2000
  95 FOR F=18 TO 21
 100 PRINT AT F,0;"
                                ";
 105 FOR I=1 TO 5
 107 NEXT I
 110 PRINT AT F,0;"
                                ";
 115 NEXT F
 117 GOTO 55
 120 GOSUB 9000
 160 PRINT AT 21,0;"HIT G FOR AN
OTHER,S TO STOP";
 170 LET D$=INKEY$
 175 IF D$<>"G" THEN IF D$<>"S"
THEN GOTO 170
 180 IF D$="G" THEN GOTO 1
 499 STOP
 500 PRINT AT 19,0;"YOUR GO. TAK
E THREE SHOTS              ";
 501 FOR V=1 TO 3
 505 LET PLR=PLR+1
 513 PRINT AT 21,0;"
                                ";
 515 PRINT AT 21,0;"SHOT NO. ";P
LR;"..FEED ROW";
 520 INPUT A
 521 IF A=0 THEN LET A=10
 522 IF A>10 THEN GOTO 515
 523 PRINT AT 21,0;"
                                ";
 525 PRINT AT 21,0;"SHOT NO. ";P
LR;"..FEED COL";
 527 INPUT B
 530 IF B>10 THEN GOTO 525
 531 IF B=0 THEN LET B=10
 532 IF A$(A,B)="X" THEN GOTO 51
1
 533 PRINT AT 21,0;"
                                ";
 535 IF A$(A,B)="*" THEN PRINT A
T 21,25;"MISSED";
 536 IF A$(A,B)<>"*" THEN IF A$(
A,B)<>"X" THEN LET HITP=HITP+1
 537 IF A$(A,B)<>"*" THEN IF A$(
A,B)<>"X" THEN PRINT AT 21,25;"
 HIT  ";
 538 FOR P=1 TO 20
 539 NEXT P
 540 IF A$(A,B)="B" THEN PRINT A
T 2+A,19+B;"B";
 545 IF A$(A,B)="C" THEN PRINT A
T 2+A,19+B;"C";
 550 IF A$(A,B)="D" THEN PRINT A
T 2+A,19+B;"D";
 555 IF A$(A,B)="S" THEN PRINT A
T 2+A,19+B;"S";
 558 IF A$(A,B)="*" THEN PRINT A
T 2+A,19+B;"X";
 560 LET A$(A,B)="X"
 565 IF HITP=20 THEN GOTO 120
 600 NEXT V
 601 RETURN
1000 REM
1001 REM  PLAYER ENTRY
1002 REM
1015 FOR S=1 TO 4
1016 PRINT AT 17,0;"TAKE CARE, S
PACE THEM THE SAME";
1020 PRINT AT 18,0;"BATTLESHIP S
QUARE ";S;
1025 PRINT AT 19,0;"ROW..";
1027 INPUT A
1028 IF A=0 THEN LET A=10
1029 IF A>10 THEN GOTO 1027
1030 PRINT AT 19,0;"COL..";
1035 INPUT B
1036 IF B=0 THEN LET B=10
1037 IF B>10 THEN GOTO 1035
1040 LET B$(A,B)="B"
1045 PRINT AT 2+A,1+B;"B";
1050 NEXT S
1053 FOR M=1 TO 2
1054 FOR N=1 TO 3
1055 PRINT AT 18,0;"CRUISER NUMB
ER ";M;" SQUARE ";N;
1058 PRINT AT 19,0;"ROW..";
1060 INPUT A
1062 IF A>10 THEN GOTO 1060
1063 IF A=0 THEN LET A=10
1064 PRINT AT 19,0;"COL..";
1066 INPUT B
1067 IF B=0 THEN LET B=10
1068 IF B>10 THEN GOTO 1066
1070 LET B$(A,B)="C"
1075 PRINT AT 2+A,1+B;"C";
1077 NEXT N
1080 NEXT M
1090 FOR M=1 TO 3
1095 FOR N=1 TO 2
1100 PRINT AT 18,0;"DESTROYER NO
.";M;" SQUARE ";N;"    "
1110 PRINT AT 19,0;"ROW..";
1120 INPUT A
1121 IF A=0 THEN LET A=10
1122 IF A>10 THEN GOTO 1120
1125 PRINT AT 19,0;"COL..";
1130 INPUT B
```



```
1131 IF B=0 THEN LET B=10
1132 IF B>10 THEN GOTO 1130
1135 LET B$(A,B)="D"
1140 PRINT AT 2+A,1+B;"[illegible]";
1150 NEXT N
1160 NEXT M
1170 FOR S=1 TO 4
1175 PRINT AT 18,0;"SUBMARINE NU
MBER..";S;"
                    ";
1180 PRINT AT 19,0;"[illegible]";
1190 INPUT A
1191 IF A=0 THEN LET A=10
1192 IF A>10 THEN GOTO 1190
1195 PRINT AT 19,0;"[illegible]";
1197 INPUT B
1198 IF B=0 THEN LET B=10
1199 IF B>10 THEN GOTO 1190
1200 LET B$(A,B)="S"
1205 PRINT AT 2+A,1+B;"[illegible]";
1210 NEXT S
1220 FOR A=17 TO 21
1225 PRINT AT A,0;"
                    ";
1230 NEXT A
1250 RETURN
2000 REM ████████████████
2001 REM █MACHINE FIRES BACK█
2002 REM ████████████████
2010 PRINT AT 19,0;"[illegible]
[illegible]";
2011 FOR T=1 TO 3
2022 LET MCN=MCN+1
2025 LET ZX=0
2040 LET A=INT (RND*10)+1
2050 LET B=INT (RND*10)+1
2060 IF B$(A,B)<>"*" THEN IF B$(
A,B)="X" THEN GOTO 2040
2063 PRINT AT 21,0;"
                    ";
2065 PRINT AT 21,0;"SHOT NO.";MC
N;" IS..";A;",";B;
2070 IF B$(A,B)="S" OR B$(A,B)="
" OR B$(A,B)="C" OR B$(A,B)="D"
 THEN GOSUB 3000
2085 IF ZX=1 THEN GOTO 2125
2110 PRINT AT 18,0;"[illegible]
[illegible]";
2112 FOR U=1 TO 20
2115 NEXT U
2117 PRINT AT 18,0;"
                    ";
2120 LET B$(A,B)="X"
2121 PRINT AT 2+A,1+B;"X";
2125 NEXT T
2999 RETURN
3000 REM ████████████████
3001 REM █MAIN FLEET HITS█
3002 REM ████████████████
3010 PRINT AT 2+A,1+B;"[illegible]";
3020 LET B$(A,B)="X"
3021 LET AA=A
3022 IF A=10 THEN LET AA=9
3030 IF B$(AA+1,B)="*" THEN LET
B$(AA+1,B)="X"
3031 IF AA+2>10 THEN GOTO 3039
3032 IF B$(AA+2,B)="*" THEN LET
B$(AA+2,B)="X"
3039 LET AA=A
3040 IF A=1 THEN LET AA=2
3050 IF B$(AA-1,B)="*" THEN LET
B$(AA-1,B)="X"
3055 IF AA-2<1 THEN GOTO 3060
3057 IF B$(AA-2,B)="*" THEN LET
B$(AA-2,B)="X"
3060 LET BB=B
3070 IF B=10 THEN LET BB=9
3080 IF B$(A,BB+1)="*" THEN LET
B$(A,B+1)="X"
3085 IF BB+2>10 THEN GOTO 3090
3087 IF B$(A,BB+2)="*" THEN LET
B$(A,BB+2)="X"
3090 LET BB=B
3100 IF B=1 THEN LET BB=2
3110 IF B$(A,BB-1)="*" THEN LET
B$(A,BB-1)="X"
3120 PRINT AT 18,0;"[illegible]
[illegible]";
3130 LET ZX=1
3140 LET HITM=HITM+1
3145 IF HITM=20 THEN GOTO 120
3150 FOR U=1 TO 30
3160 NEXT U
3170 PRINT AT 18,0;"
                    ";
3180 RETURN
5000 LET M$="████████████████
████████████"
5010 LET Z$=M$
5020 LET N$="█1234567890████████
█1234567890█"
5030 LET Y$=N$
5040 LET L$="████████████████
████████████"
5045 LET K$=L$
5050 LET O$="█1**********1██████
1**********1█"
5060 LET P$="█2**********2██████
2**********2█"
5070 LET Q$="█3**********3██████
3**********3█"
5080 LET R$="█4**********4██████
4**********4█"
5090 LET S$="█5**********5██████
5**********5█"
5100 LET T$="█6**********6██████
6**********6█"
5110 LET U$="█7**********7██████
7**********7█"
5120 LET V$="█8**********8██████
8**********8█"
5130 LET W$="█9**********9██████
9**********9█"
5140 LET X$="█0**********0██████
0**********0█"
5150 RETURN
5500 REM ██████████████████████
██
5501 REM █SET THE MACHINES PIECE
S█
5502 REM ██████████████████████
██
5510 DIM A$(10,10)
5511 DIM B$(10,10)
5512 DIM H$(10,10)
5518 REM ████████████
5519 REM █BATTLESHIP█
5520 REM ████████████
5521 FOR Y=1 TO 10
5522 FOR Z=1 TO 10
5523 LET A$(Y,Z)="*"
5524 LET B$(Y,Z)="*"
5525 NEXT Z
5526 NEXT Y
5530 LET A=INT (RND*10)+1
5531 IF A<4 THEN GOTO 5530
5540 LET B=INT (RND*10)+1
5541 IF B<4 THEN GOTO 5540
5550 IF A>7 THEN GOTO 5700
5560 IF B>7 THEN GOTO 5700
5565 IF A>B THEN GOTO 5600
5567 FOR I=1 TO 4
5568 LET A$(A,B-1+I)="B"
5569 NEXT I
5570 GOTO 5730
5600 FOR I=1 TO 4
5601 LET A$(A-1+I,B)="B"
5602 NEXT I
5605 GOTO 5730
5700 REM ██████████████████████
██
5701 REM █SETTING B$HIP AT BOTTO
M█
5702 REM ██████████████████████
██
5705 IF A>B THEN GOTO 5720
5707 FOR I=1 TO 4
5710 LET A$(A,B+1-I)="B"
5711 NEXT I
5715 GOTO 5730
5720 FOR I=1 TO 4
5721 LET A$(A+1-I,B)="B"
5722 NEXT I
5728 REM ███████████████████
5729 REM █BATTLESHIP FINISHED█
5730 REM ███████████████████
5732 FOR J=1 TO 2
5735 LET A=INT (RND*10)+1
5736 IF A<3 THEN GOTO 5735
5737 LET B=INT (RND*10)+1
5738 IF B<3 THEN GOTO 5737
5740 IF A$(A,B)<>"*" THEN GOTO 5
735
```

```
5742 IF A>8 THEN GOTO 5800
5743 IF B>8 THEN GOTO 5800
5744 IF A>B THEN GOTO 5780
5745 FOR I=1 TO 3
5747 IF A$(A,B-1+I)<>"*" THEN GO
TO 5735
5748 NEXT I
5749 FOR I=1 TO 3
5750 LET A$(A,B-1+I)="C"
5751 NEXT I
5752 GOTO 5900
5780 FOR I=1 TO 3
5782 NEXT I
5783 FOR I=1 TO 3
5784 LET A$(A-1+I,B)="C"
5785 NEXT I
5786 GOTO 5900
5800 IF A>B THEN GOTO 5850
5802 FOR I=1 TO 3
5803 IF A$(A,B+1-I)<>"*" THEN GO
TO 5735
5804 NEXT I
5805 FOR I=1 TO 3
5806 LET A$(A,B+1-I)="C"
5807 NEXT I
5810 GOTO 5900
5850 FOR I=1 TO 3
5851 IF A$(A+1-I,B)<>"*" THEN GO
TO 5735
5852 NEXT I
5853 FOR I=1 TO 3
5854 LET A$(A+1-I,B)="C"
5855 NEXT I
5900 NEXT J
5908 REM
5909 REM DESTROYERS NEXT
5910 REM
5911 FOR J=1 TO 3
5912 LET A=INT (RND*10)+1
5913 IF A<2 THEN GOTO 5912
5914 LET B=INT (RND*10)+1
5915 IF B<2 THEN GOTO 5914
5916 IF A$(A,B)<>"*" THEN GOTO 5
912
5920 IF A>9 THEN GOTO 5970
5921 IF B>9 THEN GOTO 5970
5922 IF A>B THEN GOTO 5950
5925 FOR I=1 TO 2
5926 IF A$(A,B-1+I)<>"*" THEN GO
TO 5912
5927 NEXT I
5930 FOR I=1 TO 2
5931 LET A$(A,B-1+I)="D"
5932 NEXT I
5933 GOTO 6000
5950 FOR I=1 TO 2
5951 IF A$(A-1+I,B)<>"*" THEN GO
TO 5912
5952 NEXT I
5953 FOR I=1 TO 2
5954 LET A$(A-1+I,B)="D"
5959 NEXT I
5960 GOTO 6000
5970 IF A>B THEN GOTO 5990
5971 FOR I=1 TO 2
5972 IF A$(A,B+1-I)<>"*" THEN GO
TO 5912
5973 NEXT I
5974 FOR I=1 TO 2
5975 LET A$(A,B+1-I)="D"
5976 NEXT I
5979 GOTO 6000
5990 FOR I=1 TO 2
5991 IF A$(A+1-I,B)<>"*" THEN GO
TO 5912
5992 NEXT I
5993 FOR I=1 TO 2
5994 LET A$(A+1-I,B)="D"
5995 NEXT I
6000 NEXT J
6100 FOR J=1 TO 4
6105 LET A=(INT (RND*10))+1
6107 LET B=(INT (RND*10))+1
6110 IF A$(A,B)<>"*" THEN GOTO 6
105
6115 LET A$(A,B)="S"
6120 NEXT J
6125 FOR X=1 TO 10
6126 FOR Y=1 TO 10
6127 LET H$(X,Y)=A$(X,Y)
6128 NEXT Y
6129 NEXT X
5999 RETURN
7000 SLOW
7008 REM
7009 REM PRINT SCREEN
7010 REM
7015 PRINT K$;M$;N$;O$;P$;Q$;R$;
S$;T$;U$;V$;W$;X$;Y$;Z$;L$;
7020 RETURN
7500 REM
7501 REM INITIAL DISPLAY
7502 REM
7510 PRINT AT 2,9;"████████████
██";
7511 PRINT AT 3,9;"█
█";
7520 PRINT AT 4,9;"█ BATTLESHIPS
█";
7522 PRINT AT 5,9;"█
█";
7525 PRINT AT 6,9;"████████████
██";
7530 PRINT AT 15,3;"DO YOU WANT
INSTRUCTIONS?"
7535 PRINT AT 16,10;"( Y OR N )"
7540 LET Q$=INKEY$
7550 IF Q$<>"Y" THEN IF Q$<>"N"
THEN GOTO 7540
7560 IF Q$="Y" THEN GOSUB 8000
7565 CLS
7600 RETURN
8000 CLS
8010 PRINT "   THE OBJECT IS TO
SINK THE "
8012 PRINT "MACHINES FLEET BEFOR
E IT SINKS  YOURS."
8015 PRINT "   THE MACHINE FIRST
 OF ALL     POSITIONS ITS OWN FL
EET."
8020 PRINT "   YOU ARE THEN ASKE
D TO INPUT  YOUR BATTLESHIPS."
8022 PRINT
8025 PRINT "ONE BATTLESHIP (4 SQ
UARES), TWO"
8030 PRINT "CRUISERS (3 SQUARES)
, THREE"
8035 PRINT "DESTROYERS (2 SQUARE
S) AND FOUR"
8040 PRINT "SUBMARINES (1 SQUARE
 EACH)."
8045 PRINT "   YOU THEN FIRE 3 S
HOTS, THE"
8050 PRINT "MACHINE HAS 3, AND S
O ON."
8055 PRINT "   LOWEST NUMBER OF
SHOTS WINS."
8060 PRINT
8101 PRINT "THE MACHINE DOES NOT
 THINK AS "
8102 PRINT "YOU DO, BUT IT HAS S
PIES TO GIVE"
8103 PRINT "IT INFORMATION ABOUT
 YOUR FLEET."
8104 PRINT "IT GETS THIS AFTER E
ACH HIT"
8106 PRINT AT 21,8;"(PLEASE WAIT
)"
8110 FOR G=1 TO 200
8120 NEXT G
8130 CLS
8140 RETURN
9000 FOR U=1 TO 20
9010 FAST
9012 FOR I=1 TO 5
9015 NEXT I
9029 SLOW
9030 NEXT U
9040 IF HITP=20 THEN PRINT AT 17
,0;"WELL DONE YOU HAVE WON
"
9050 IF HITP=20 THEN GOTO 9300
9060 PRINT AT 17,0;"I HAVE BEATE
N YOU.";
9070 PRINT AT 18,0;"HERE IS MY W
HOLE FLEET";
9080 FOR A=1 TO 10
9090 FOR B=1 TO 10
9100 PRINT AT 2+A,19+B;H$(A,B);
9110 NEXT B
9120 NEXT A
9300 RETURN
```

Chris Cox

HOUSE OF HORROR

Dare you enter the haunted house?

It was a dark and stormy night (as Victorian writer, Edward Bulwer Lytton, used to say) when Chris Cox decided to write this program!

The plot of the game is that you have been transported to a deserted mansion, full of hidden treasure. All of the treasure was kindly left by the last owner of the house, but unfortunately he also left a number of his 'pets' to guard his fortune. And yes, you've guessed it, his 'pets' (in the shape of monsters!) are none too pleased by your intrusion. And if that wasn't enough, you have to watch out for the roaming reaper who has a liking for people's heads — it's not too interested in the rest of your body though (need I say more!).

GO EXPLORING

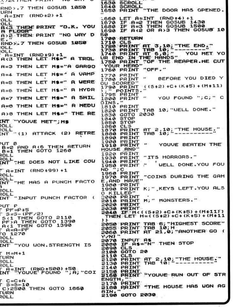
Instructions appear throughout the program telling you in which direction you can go. You move around the randomly-generated selection of halls and stairways constantly exploring until you find the treasure, keys to open some of the doors and, of course, a grim selection of monsters.

The more coins you find, the less frequent large finds become; however, as you come across less coins, you'll find more keys and have more monsters which means you can explore more of the house.

The game ends when you have found over 2,500 coins, run out of strength warding off the monsters or you've had your head removed by the roaming reaper!

Happy hunting!

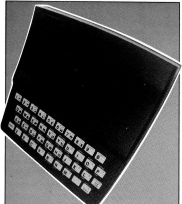
```
  10 LET H=0
  20 LET S=200
  30 LET C=0
  40 LET K=0
  50 LET M=0
  55 PRINT AT 3,9;"""THE HOUSE""
"
  60 PRINT AT 8,4;"INSTRUCTIONS 
FOLLOW"
  70 PAUSE 200
  80 CLS
  90 PRINT AT 2,10;"THE HOUSE."
 100 PRINT TAB 10;"----------"
 110 PRINT
 120 PRINT
 130 PRINT "   YOU HAVE BEEN TRA
NSPORTED TO"
 140 PRINT "A DESERTED MANSION.A
LL AROUND IT"
 150 PRINT "TRESURE IS HIDDEN.IT
 WAS LEFT"
 160 PRINT "BY THE LAST OWNER."
 170 PRINT
 180 PRINT "AS WELL AS HIS TREAS
URE HE LEFT"
 190 PRINT "HIS PET MONSTERS TO
GUARD IT."
 200 PRINT "BEWARE THE CLUTCHES
OF THE ROAM-"
 210 PRINT "ING REAPER.........
............"
 220 PRINT AT 21,0;"PRESS ANY KE
Y TO START."
 230 PAUSE 3000
 240 CLS
 250 LET A$=" A HALL."
 260 LET B$=" A DOOR."
 270 LET C$=" A STAIRWAY."
 280 LET A=INT (RND*3)+1
 290 IF A=1 THEN LET X$=C$
 300 IF A=2 THEN LET X$=A$
 310 IF A=3 THEN LET X$=B$
 320 IF A>3 THEN LET X$=A$
 330 PRINT AT 21,0;"YOU ARE FACI
NG";X$
 340 SCROLL
 350 SCROLL
 360 IF X$=A$ THEN GOSUB 420
 370 IF X$=B$ THEN GOSUB 480
 380 IF X$=C$ THEN GOSUB 530
 390 SCROLL
 400 SCROLL
 410 GOTO 250
 420 PRINT "(1) MOVE (2) EXPLORE
(3) STATUS"
 430 INPUT B
 440 IF B=1 THEN GOSUB 590
 450 IF B=2 THEN GOSUB 640
 460 IF B=3 THEN GOSUB 710
 470 RETURN
 480 PRINT "(1) OPEN (2) LEAVE"
 490 INPUT Z
 500 IF Z=1 THEN GOSUB 870
 510 IF Z=2 THEN RETURN
 520 RETURN
 530 PRINT "(1) UP (2) DOWN (3)
LEAVE"
 540 INPUT D
 550 IF D=1 THEN GOSUB 910
 560 IF D=2 THEN GOSUB 980
 570 IF D=3 THEN RETURN
 580 RETURN
 590 SCROLL
 600 SCROLL
 610 PRINT "(1)NORTH(2)SOUTH(3)E
AST(4)WEST"
 620 INPUT X
 630 RETURN
 640 SCROLL
 650 SCROLL
 660 LET A=INT (RND*3)+1
 670 IF A>2 THEN GOSUB 1520
 680 IF A=1 THEN GOSUB 1070
 690 IF A=2 THEN GOSUB 1450
 700 RETURN
 710 CLS
 720 PRINT AT 3,10;"STATUS"
 730 PRINT
 740 PRINT
 750 PRINT TAB 4;"STRENGTH
";S
 760 PRINT
 770 PRINT TAB 4;"COINS
";C
 780 PRINT
 790 PRINT TAB 4;"KEYS
";K
 800 PRINT
 810 PRINT TAB 4;"MONSTERS DEAD
";M
 820 PRINT AT 15,6;"SCORE ";((S*
2)+C+(K*5)+(M+11))
```



```
 830 PRINT AT 21,0;"PRESS ANY KE
Y TO CONTINUE......"
 840 PAUSE 3000
 850 CLS
 860 RETURN
 870 LET X=INT (RND*4)+1
 880 IF X<3 THEN GOSUB 1550
 890 IF X>2 THEN GOSUB 1630
 900 RETURN
 910 LET X=INT (RND*2)+1
 920 SCROLL
 930 SCROLL
 940 IF X<3 THEN PRINT "O.K. YOU
RE UP A FLOOR"
 950 IF X>2 THEN PRINT "NO WAY U
P THERE."
 960 IF RND>.7 THEN GOSUB 1050
 970 RETURN
 980 LET A=INT (RND*2)+1
 990 SCROLL
1000 SCROLL
1010 IF A<3 THEN PRINT "O.K. YOU
RE DOWN A FLOOR"
1020 IF A>2 THEN PRINT "NO WAY D
OWN THERE."
1030 IF RND>.7 THEN GOSUB 1050
1050 SCROLL
1060 SCROLL
1070 LET A=INT (RND*9)+1
1080 IF A<3 THEN LET M$=" A TROL
L"
1090 IF A=3 THEN LET M$="A GARGO
YLE"
1100 IF A=4 THEN LET M$=" A VAMP
IRE."
1110 IF A=5 THEN LET M$=" A WERE
WOLF."
1120 IF A=6 THEN LET M$=" A HYDR
A."
1130 IF A=7 THEN LET M$=" A SHIL
O."
1145 IF A=8 THEN LET M$=" A MEDU
SA."
1150 IF A>8 THEN LET M$=" THE RE
APER."
1170 PRINT "YOUVE MET";M$
1180 SCROLL
1185 SCROLL
1190 PRINT "(1) ATTACK (2) RETRE
AT"
1200 INPUT B
1210 IF B=2 AND A<6 THEN RETURN
1220 IF B=1 THEN GOTO 1260
1230 SCROLL
1240 SCROLL
1250 PRINT "HE DOES NOT LIKE COW
ARDS."
1260 LET A=INT (RND*99)+1
1270 SCROLL
1280 SCROLL
1290 PRINT "HE HAS A PUNCH FACTO
R OF ";A
1300 SCROLL
1310 SCROLL
1320 PRINT "INPUT PUNCH FACTOR (
1-9)"
1330 INPUT P
1340 LET PF=P*5
1350 LET S=S-(PF/2)
1355 IF S<1 THEN GOTO 2110
1360 IF PF>A THEN GOTO 1390
1365 IF A<1 THEN GOTO 1390
1370 LET A=A-PF
1380 GOTO 1270
1390 SCROLL
1400 SCROLL
1410 PRINT "YOU WON.STRENGTH IS
NOW ";S
1415 LET M=M+1
1420 RETURN
1430 SCROLL
1440 SCROLL
1450 LET A=INT (RND*500)+50
1460 PRINT "YOUVE FOUND ";A;" COI
NS."
1470 LET C=C+A
1475 LET S=S-10
1480 IF C>2500 THEN GOTO 1860
1490 RETURN
1500 SCROLL
1510 SCROLL
1520 PRINT "YOUVE FOUND A KEY."
1530 LET K=K+1
1535 LET S=S+20
1540 RETURN
1550 SCROLL
1560 SCROLL
1570 PRINT "DOOR LOCKED.GOT A KE
Y ?"
1580 IF K<1 THEN RETURN
1590 SCROLL
1600 SCROLL
1610 PRINT "YES."
1620 LET K=K-1
1630 SCROLL
1640 SCROLL
1650 PRINT "THE DOOR HAS OPENED.
"
1660 LET A=INT (RND*4)+1
1670 IF A=2 THEN GOSUB 1430
1680 IF A=3 THEN GOSUB 1500
1690 IF A<2 OR A>3 THEN GOSUB 10
50
1700 RETURN
1710 CLS
1720 PRINT AT 3,10;"THE END."
1730 PRINT TAB 10;"--------"
1740 PRINT AT 6,0;"   YOU MET YO
UR END IN THE HANDS"
1750 PRINT "OF THE REAPER.HE CUT
 YOUR HEAD"
1760 PRINT "OFF."
1770 PRINT
1780 PRINT "   BEFORE YOU DIED Y
OU SCORED"
1790 PRINT ((S*2)+C+(K*5)+(M*11)
);" POINTS."
1795 PRINT
1800 PRINT "   YOU FOUND ";C;" C
OINS."
1810 PRINT
1820 PRINT TAB 10;"WELL DONE."
1830 GOTO 2030
1840 STOP
1850 STOP
1860 CLS
1870 PRINT AT 2,10;"THE HOUSE."
1880 PRINT TAB 10;"----------"
1890 PRINT
1900 PRINT
1910 PRINT "   YOUVE BEATEN THE
HOUSE AND"
1920 PRINT
1930 PRINT "ITS HORRORS."
1940 PRINT
1950 PRINT "   WELL DONE.YOU FOU
ND ";C
1960 PRINT
1970 PRINT "COINS DURING THE GAM
E,AND HAVE"
1980 PRINT
1990 PRINT K;" KEYS LEFT.YOU ALS
O KILLED"
2000 PRINT
2010 PRINT M;" MONSTERS."
2020 PRINT
2030 PRINT
2040 IF H<((S*2)+C+(K*5)+(M*11))
 THEN LET H=((S*2)+C+(K*5)+(M*11
))
2050 PRINT TAB 6;"HIGHEST SCORE"
2055 PRINT TAB 10;H
2060 PRINT AT 21,0;"ANOTHER GO (
Y/N)"
2070 INPUT A$
2080 IF A$="N" THEN STOP
2090 CLS
2100 GOTO 20
2110 CLS
2120 PRINT AT 2,10;"THE HOUSE."
2130 PRINT TAB 10:"----------"
2140 PRINT
2150 PRINT
2160 PRINT "YOUVE RUN OUT OF STR
ENGTH."
2170 PRINT
2180 PRINT "THE HOUSE HAS WON AG
AIN."
2190 GOTO 2030
```

S P Stratford

DROP OUT



A ZX81 version of the popular arcade game.

This is a game for the ZX81 with at least 3½K of RAM and has been written so that, apart from the enjoyment I hope you'll get from the game, you will also learn more about the capabilities of the computer.

The idea of the game is to drop your arrow, a 'V', into the mouth of a bottle. If you miss the mouth of the bottle or land in one of the other bottles, you score the amount of points indicated at the bottom of the bottle (printed in inverse print). You have ten attempts to score as many points as possible, and you lose one of your goes for each 'V' dropped. To drop a 'V', press any key apart from Break or Shift.

GOTTA LOTTA BOTTLE

When the program is RUN, the bottles appear and your '>' moves rapidly across the screen. Pressing a key will make your arrow fall from its path, hopefully into the mouth of a bottle. When your ten attempts have been exhausted, your score and the highest score so far will be displayed.

The program uses the memory mapped display of the ZX81 with at least 3½K. I have deliberately not included instructions in my program so that users with more than 1K but less than 16K can also enjoy the game.

I have used variables to represent areas of the program, for example:

```
70 GOSUB DRAWG
```

This line makes the program go

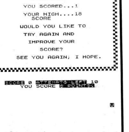
PROGRAM VARIABLES

5	— Initialises the main variables used in the program.
10	— Finds the beginning of the display file.
20	— The starting position of your arrow character.
210-230	— The routine to find out if your arrow has hit anything.
280	— The routine to print on the screen how much you scored.
710	— Prints your points on the screen.

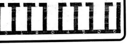
to the subroutine to print the bottles.
Here follows a brief explanation of some of the more important lines of the program.

If the user wishes to have a delay before the game starts, he or she should add the line:
7 PAUSE 4E4
Then, by pressing a key, the game will start. When the program has been typed in, RUN 9000 will SAVE the program to tape and will then RUN the main program.

```
   5 GOSUB 9000
   6 REM **R.A.M. USED: 3.5K**
  10 LET DFILE=PEEK 16396+256*PE
EK 16397+1
  20 LET HM=DFILE+198
  30 LET HI=0
  40 LET ATT=10
  50 LET S=0
  60 CLS
  70 GOSUB DRAWG
  80 FOR X=0 TO 31
  90 POKE HM+X,18
 110 IF INKEY$<>"" THEN GOTO DRO
P
 120 POKE HM+X,0
 130 NEXT X
 140 GOTO 80
 199 REM **DROP**
 200 POKE HM+X,187
 210 FOR Y=1 TO 16
 220 IF PEEK (HM+X+(Y*33))=128 T
HEN GOTO 270
 230 IF PEEK (HM+X+(Y*33))<>0 TH
EN GOTO SCORE
 240 POKE HM+X+(Y*33),59
 250 POKE HM+X+(Y*33),0
 260 NEXT Y
 269 REM **NO SCORE**
 270 LET B$="NO SCORE."
 280 GOSUB DISPLAY
 290 LET ATT=ATT-1
 300 IF ATT=0 THEN GOTO END
 310 GOTO 60
 399 REM **SCORE**
 400 LET SC=PEEK (HM+X+(Y*33))
 410 LET SC=SC-156
 420 GOSUB 430+(10*SC)
 425 LET S=S+SC
 427 GOTO 280
 430 GOTO 270
 440 LET B$="YOU SCORE 1 POINT."
 445 RETURN
 450 LET B$="YOU SCORE 2 POINTS.
"
 455 RETURN
 460 LET B$="YOU SCORE 3 POINTS.
"
 465 RETURN
 470 LET B$="YOU SCORE 4 POINTS.
"
 475 RETURN
 480 LET B$="YOU SCORE 5 POINTS.
"
 485 RETURN
 490 LET B$="YOU SCORE 6 POINTS.
"
 495 RETURN
 499 REM **END**
 500 LET B$="▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚▚
▚▚▚▚▚▚▚"
 510 LET C$=B$(1)+"
                              "+B$(1)
 520 CLS
 530 PRINT AT 0,0;B$;AT 21,0;B$
 540 FOR N=1 TO 20
 550 PRINT AT N,0;C$
 560 NEXT N
 570 IF HI<S THEN LET HI=S
 580 PRINT AT 2,10;"DROP-OUT";AT
 3,10;"▀▀▀▀▀▀▀▀"
 590 PRINT AT 5,8;"YOU SCORED...
";S
 600 PRINT AT 7,8;"YOUR HIGH....
";HI;TAB 10;"SCORE"
 610 PRINT AT 10,7;"WOULD YOU LI
KE TO";AT 12,8;"TRY AGAIN AND";A
T 14,9;"IMPROVE YOUR";AT 16,12;"
SCORE?"
 620 IF INKEY$="N" THEN GOTO 650
 630 IF INKEY$="" THEN GOTO 620
 640 GOTO 40
 650 PRINT AT 18,6;"SEE YOU AGAI
N, I HOPE."
 660 STOP
 699 REM **DISPLAY**
 700 LET P$="
               "
 710 PRINT AT 1,((32-LEN B$)/2)
B$
 720 FOR I=1 TO 100
 730 NEXT I
 740 PRINT AT 1,0;P$
 750 RETURN
 799 REM **DRAWG**
 800 PRINT AT 0,3;"SCORE ";S;" A
TTEMPTS LEFT ";ATT
 810 IF ATT<10 THEN PRINT AT 0,3
1;" "
 820 PRINT AT 14,0;
 830 PRINT "██ ███ ████ ███  ██ █
██ ██   ██ █"
 840 LET U$="█   █   █   █   █
█   █   █   █"
 850 FOR H=15 TO 20
 860 PRINT AT H,0;U$
 870 NEXT H
 880 PRINT "███3███1███2███0███4
███5███0███6█"
 890 RETURN
8999 REM **INITIALISE**
9000 LET DRAWG=800
9010 LET DISPLAY=900
9020 LET END=500
9030 LET DROP=200
9040 LET SCORE=400
9100 RETURN
9200 SAVE "DROP-OUT"
9210 RUN 5
```



```
DROP-OUT

YOU SCORED...1

YOUR HIGH....18
  SCORE

WOULD YOU LIKE TO

TRY AGAIN AND

IMPROVE YOUR

SCORE?

SEE YOU AGAIN; I HOPE.
```



```
SCORE 0 ATTEMPTS LEFT 10
    YOU SCORE 6 POINTS.
```

Nick Brown

MATHS MAZE

A great game with an educational twist.

At the start of the game you must select a level of difficulty, from one to five, five being the hardest and almost impossible. The object of the game is to move around the maze, you being the inverse quotes, using the arrow keys, '5' to '8', until you find one of the signs: '+','−', '*' or '/'. When you find one of these signs, you simply 'eat' it. The computer will then clear the screen, and set you five questions, their hardness depending on the level of difficulty selected at the start of the game. After these five questions, you are given a 'percentage rating', depending on how many sums you answered correctly. After this, press any key and the faithful ZX81 will put you back in the maze to carry on the game.

ON LEAVE

You may leave the maze at any time, but obviously the idea is to get as many points as possible. You get one point for each correctly answered addition or subtraction question, and two points for each multiplication or divide question.

At the end of each game (when you have reached the exit) you are told your score and given three options: to play again at the same level, to stop, or to play again at a different level.

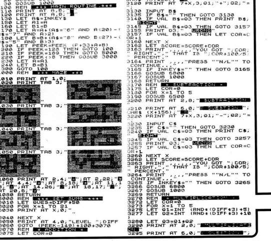
Your movement around the maze is quite fast for a BASIC program; this is due to the shortness of the loop (lines 100 to 300) and because the main work is done by subroutines.

Type in the program as published, then before RUNning the game, type the following:

POKE 16510,0

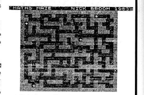
PROGRAM DESCRIPTION

Lines 100-300	Main routine.
Lines 1000-1080	Print the maze.
Lines 3000-3060	Sets the sums, deciding which sums.
Lines 3070-3168	Set the addition sums.
Lines 3170-3269	Set the subtraction sums.
Lines 3270-3369	Set the multiplication sums.
Lines 3370-3800	Set the division sums.
Lines 5000-5180	End of game routine.
Lines 6500-6600	Set the values for the sums.
Lines 6800-6840	Clear the screen, leaving the top line.
Lines 8000-8500	Variables routine.
Lines 8600-8900	Introduction, also set level of difficulty.
Lines 9000-9020	Load. The program will automatically run on subsequent loadings if the program is SAVEd using GOTO 9000.

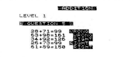


```
   1 PRINT AT 0,0;" MATHS MAZE   
 NICK BROOM 1983."
  10 GOSUB 8600
  20 GOSUB 8000
  30 GOSUB 1000
 100 REM  *** MAIN ROUTINE ***
 110 PRINT AT A,B;" "
 120 PRINT AT A,B;"█"
 130 LET A$=INKEY$
 150 LET A1=A
 160 LET B1=B
 170 LET A=A+(A$="6" AND A<20)-(
A$="7" AND A>2)
 180 LET B=B+(A$="8" AND B<27)-(
A$="5" AND B>3)
 190 LET PEEK=PEEK (F+33*A+B)
 200 IF PEEK=128 THEN GOTO 100
 210 IF PEEK=189 THEN GOTO 5000
 220 IF PEEK<>8 THEN GOSUB 3000
 230 LET A=A1
 240 LET B=B1
 300 GOTO 100
1000 REM  *** MAZE ***
1010 PRINT AT 1,0;
1020 PRINT TAB 3;"
"
1030 PRINT TAB 3;"
"
1040 PRINT TAB 3;"
"
1050 PRINT TAB 3;"
"
1060 PRINT AT 2,4;"█";AT 2,22;"█
";AT 4,4;"█";AT 8,18;"█";AT 11,1
8;"█";AT 14,26;"█";AT 18,17;"█";
AT 20,6;"█"
1080 RETURN
3000 REM  *** SET SUMS ***
3010 LET QUES=DIFF*50
3020 FOR X=1 TO 21
3030 PRINT AT X,0;"
                               "
3040 NEXT X
3050 PRINT AT 4,0;"LEVEL ";DIFF
3060 GOTO (PEEK-149)*100+3070
3070 REM  * ADDITION *
3075 LET COR=0
3080 FOR X=1 TO 5
3090 GOSUB 6500
3100 PRINT AT 2,10;" ADDITION "
3110 PRINT AT 6,0;"  QUESTION ";
CHR$ (X+156);"  "
3120 PRINT AT 7+X,3;Q1;"+";Q2;"=
";
3130 INPUT B$
3135 IF B$="" THEN GOTO 3130
3140 IF VAL B$=Q3 THEN PRINT B$;
"   RIGHT"
3145 IF VAL B$=Q3 THEN GOTO 3157
3155 PRINT Q3;"     WRONG"
3157 IF VAL B$=Q3 THEN LET COR=C
OR+1
3160 NEXT X
3162 LET SCORE=SCORE+COR
3163 PRINT ,,,,,,"YOU GOT ";COR;
" RIGHT."," THAT IS ";COR*100/5;
" PERCENT."
3164 PRINT ,,,,"PRESS ""N/L"" TO
 CONTINUE..."
3165 IF INKEY$="" THEN GOTO 3165
3166 GOSUB 6800
3167 GOSUB 1000
3168 RETURN
3170 REM  * SUBTRACTION -
3175 LET COR=0
3180 FOR X=1 TO 5
3190 GOSUB 6500
3200 PRINT AT 2,8;" SUBTRACTION 
"
3210 PRINT AT 6,0;"  QUESTION ";
CHR$ (X+156);"  "
3220 PRINT AT 7+X,3;Q1;"-";Q2;"=
";
3230 INPUT C$
3235 IF C$="" THEN GOTO 3230
3240 IF VAL C$=Q3 THEN PRINT C$;
"   RIGHT"
3245 IF VAL C$=Q3 THEN GOTO 3257
3255 PRINT Q3;"     WRONG"
3257 IF VAL C$=Q3 THEN LET COR=C
OR+1
3260 NEXT X
3262 LET SCORE=SCORE+COR
3263 PRINT ,,,,,,"YOU GOT ";COR;
" RIGHT."," THAT IS ";COR*100/5;
" PERCENT."
3264 PRINT ,,,,"PRESS ""N/L"" TO
 CONTINUE..."
3265 IF INKEY$="" THEN GOTO 3265
3266 GOSUB 6800
3267 GOSUB 1000
3269 RETURN
3270 REM  * MULTIPLICATION *
3272 LET COR=0
3273 FOR X=1 TO 5
3275 LET Q1=INT (RND*(DIFF*3)+5)
3277 LET Q2=INT (RND*(DIFF*3)+10
)
3280 LET Q3=Q1*Q2
3290 PRINT AT 2,8;" MULTIPLICATI
ON "
3295 PRINT AT 6,0;"  QUESTION ";
```

```
CHR$ (X+156);"█"
3300 PRINT AT 7+X,3;01;"*";02;"=
";
3310 INPUT D$
3315 IF D$="" THEN GOTO 3310
3320 IF VAL D$=03 THEN PRINT D$;
" RIGHT"
3325 IF VAL D$=03 THEN GOTO 3350
3330 PRINT 03;"  WRONG"
3350 IF VAL D$=03 THEN LET COR=C
OR+1
3355 NEXT X
3357 LET SCORE=SCORE+COR*2
3360 PRINT ,,,,,,"YOU GOT ";COR;
" RIGHT.","  THAT IS ";COR*100/5;
" PERCENT."
3362 PRINT ,,,,"PRESS ""N/L"" TO
CONTINUE..."
3363 IF INKEY$="" THEN GOTO 3363
3365 GOSUB 6800
3368 GOSUB 1000
3369 RETURN
3370 REM * DIVISION *
3372 LET COR=0
3374 FOR X=1 TO 5
3375 LET Q1=INT (RND*(DIFF*5)+10

3377 LET Q2=INT (RND*(DIFF*5)+10

3380 LET Q3=Q1*Q2
3390 PRINT AT 2,10;"[illegible]"
3395 PRINT AT 6,0;"[illegible]";
CHR$ (X+156);"█"
3400 PRINT AT 7+X,3;Q3;"/";Q2;"=
";
3410 INPUT E$
3415 IF E$="" THEN GOTO 3410
3420 IF VAL E$=Q1 THEN PRINT E$;
" RIGHT"
3425 IF VAL E$=Q1 THEN GOTO 3435
3430 PRINT Q1;"  WRONG"
3435 IF VAL E$=Q1 THEN LET COR=C
OR+1
3440 NEXT X
3450 LET SCORE=SCORE+COR*2
3460 PRINT ,,,,,,"YOU GOT ";COR;
" RIGHT.","  THAT IS ";COR*100/5;
" PERCENT."
3462 PRINT ,,,,"PRESS ""N/L"" TO
CONTINUE..."
3463 IF INKEY$="" THEN GOTO 3463
3470 GOSUB 6800
3480 GOSUB 1000
3800 RETURN
5000 REM *** THE END ***
5010 GOSUB 6800
5020 PRINT AT 8,0;" YOU HAVE REA
CHED THE EXIT."
5030 PRINT ,," YOU SCORED ";SCOR
E;" POINTS ON LEVEL ";DIFF
5065 PRINT ,,"PRESS :",,"""P"" T
O PLAY",,"""S"" TO STOP",,"""R"""
 TO CHANGE LEVEL."
5070 IF INKEY$="" THEN GOTO 5070
5080 IF INKEY$="S" THEN STOP
5090 IF INKEY$="P" THEN GOTO 511
0
5095 IF INKEY$="R" THEN GOTO 516
0
5100 GOTO 5070
5110 GOSUB 6800
5120 LET SCORE=0
5130 LET A=10
5140 LET B=3
5150 GOTO 30
5160 GOSUB 6800
5165 PRINT AT 6,0;
5170 GOSUB 8660
5180 GOTO 20
6500 REM * SET VALUES *
6510 LET Q1=INT (RND*QUES)+20
6520 LET Q2=INT (RND*QUES)+50
6530 IF PEEK=149 THEN LET Q3=Q1+
Q2
6540 IF PEEK=150 AND Q1<Q2 THEN
GOTO 6500
6550 IF PEEK=150 THEN LET Q3=Q1-
Q2
6600 RETURN
6800 REM * CLEAR SCREEN *
6810 FOR X=1 TO 21
6820 PRINT AT X,0;"
"
6830 NEXT X
6840 RETURN
8000 REM *** VARIABLES ***
8010 LET A=10
8020 LET B=3
8030 LET P=PEEK 16396+256*PEEK 1
6397+1
8040 LET COR=0
8050 LET SCORE=0
8500 RETURN
8600 REM *** INTRO ***
8640 PRINT AT 4,0;" YOU (A ""█""
) ARE IN A MAZE AND  YOUR TASK I
S TO GET TO THE EXIT (▨). "
8650 PRINT " ON YOUR WAY YOU MUS
T PICK UP   THE SIGNS (█,█,█,█)
AND ANSWER  THE CORRESPONDING QU
ESTIONS."
8655 PRINT ,,"   TO MOVE USE:
   7",,"                      5█8",,"
   6"
8660 PRINT ,,,,"PLEASE ENTER LEV
EL OF DIFFICULTY"
8670 PRINT ,,"    FROM 1 TO 5 (1-
EASY/5-HARD)"
8680 LET A$=INKEY$
8690 IF A$="" THEN GOTO 8680
8695 IF A$="1" OR A$="2" OR A$="
3" OR A$="4" OR A$="5" THEN GOTO
 8700
8697 GOTO 8680
8700 FOR X=1 TO 18
8710 PRINT AT X,0;"
"
8720 NEXT X
8730 LET DIFF=VAL A$
8900 RETURN
9000 REM *** LOAD N GO ***
9010 SAVE "MAZE"
9020 RUN
```



A sample screen illustration.

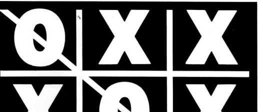


```
            ADDITION

LEVEL 1

 QUESTION 5

     28+71=99   WRONG
     63+98=161   WRONG
     34+92=126   RIGHT
     26+73=99   RIGHT
     61+89=150   RIGHT


YOU GOT 3 RIGHT.
 THAT IS 60 PERCENT.

PRESS "N/L" TO CONTINUE...
```

An example addition test.

NAUGHT BUT A GAME

The classic game of naughts and crosses for your ZX81.



This program has the ZX81 playing a pretty mean game of Naughts and Crosses against a willing opponent. The program requires about 7K of RAM to operate.

In this game you are the 'naughts' and the computer is the 'crosses'. The program is divided into several parts:

DRAWC — Draws the computer's piece.
DRAWP — Draws the player's piece.
WIN — Checks to see if someone has won or if it is a draw.
COMP — The computer's move.
PLAYER — The player's move.
CLEAR — Clears the player's prompts.
GRID — Prints the playing grid.

Each part of the program is called by the GOSUB command, for example GOSUB COMP is the subroutine for the computer's moves.

THREE IN A ROW

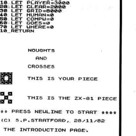
The following is a brief guide to some of the more important lines in the program:

5005-5020	Find if one of the players has a row of three counters.
5200	The player gets two points for each win.
5320	The ZX81 gets five points for each win.
6005-6008	Work out the average value of RND.
8002	Ensure the print position is in the top left hand corner.
8200-8210	Make sure that the scores do not run onto the board.

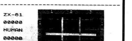
XOX

```
   1 SAVE "O"
   2 SLOW
   3 REM * 7K R.A.M. AT LEAST *
   4 RAND 0
   5 PRINT TAB 9;"NOUGHTS";AT 2,
11;"AND";AT 4,9;"CROSSES"
   6 PRINT AT 6,2;"■■■";AT 7,2;"
■ ■";TAB 9;"THIS IS YOUR PIECE";
AT 8,2;"■■■"
   7 PRINT AT 11,2;"■";TAB 4;"■"
;TAB 3;"■";TAB 9;"THIS IS THE ZX
-81 PIECE";TAB 2;"■ ■"
   8 PRINT AT 21,2;"(C) S.P.STRA
TFORD, 28/11/82"
   9 PRINT AT 16,0;"**** PRESS N
EWLINE TO START ****";AT 16,0;"*
*** PRESS NEWLINE TO START ****"
  10 IF CODE INKEY$<>118 THEN GO
TO 9
  15 GOSUB 9000
  19 CLS
  20 LET GOES=0
  21 FOR U=1 TO 3
  22 FOR P=1 TO 3
  23 LET A(P,U)=0
  24 NEXT P
  25 NEXT U
  26 GOSUB GRID
  30 PRINT AT 11,1;"DO YOU";AT 1
2,1;"WISH TO";AT 13,1;"GO FIRST?
";AT 14,2;"Y OR N."
  40 LET A$=INKEY$
  50 IF A$="N" THEN GOTO 110
  60 IF A$<>"Y" THEN GOTO 40
  90 GOSUB PLAYER
 100 GOSUB WIN
 105 GOSUB DRAWP
 110 GOSUB COMP
 115 LET GOES=GOES+1
 120 GOSUB WIN
 130 GOSUB DRAWC
 140 GOTO 90
 999 REM
1000 REM   PLAYERS PIECES
1001 REM
1010 LET U$="■■■"
1020 PRINT AT YP1-1,XP1-1;U$;AT
YP1+1,XP1-1;U$;AT YP1,XP1-1;U$(1
);AT YP1,XP1+1;U$(1)
1030 RETURN
1499 REM
1500 REM   ZX-81 PIECES
1501 REM
1510 LET U$="■"
1520 PRINT AT YC1-1,XC1-1;U$;AT
YC1-1,XC1+1;U$;AT YC1,XC1;U$;AT
YC1+1,XC1-1;U$;AT YC1+1,XC1+1;U$
1530 RETURN
1999 REM
2000 REM   CLEAR SCREEN
2001 REM
2005 LET U$="

2010 FOR Z=10 TO 21
2020 IF Z<=16 THEN PRINT AT Z,0;
U$( TO 11)
2030 IF Z>16 THEN PRINT AT Z,0;U
$
2040 NEXT Z
2050 RETURN
2999 REM
3000 REM   PLAYER MOVE
3001 REM
3010 GOSUB CLEAR
3020 PRINT AT 11,1;" TO ";AT 12,
1;"WHERE";AT 14,3;"?"
3040 LET A$=INKEY$
3050 IF A$<"1" OR A$>"3" THEN GO
TO 3040
3060 LET XP=VAL A$
3070 PRINT AT 14,3;XP;AT 16,2;"A
ND";AT 18,3;"?"
3080 LET A$=INKEY$
3090 IF A$<"1" OR A$>"3" THEN GO
TO 3080
3100 LET YP=VAL A$
3110 PRINT AT 18,3;YP
3115 IF A(XP,YP)<>0 THEN GOTO 30
10
3120 IF XP=1 THEN LET XP1=14
3130 IF XP=2 THEN LET XP1=19
3140 IF XP=3 THEN LET XP1=24
3150 IF YP=1 THEN LET YP1=3
3160 IF YP=2 THEN LET YP1=8
3170 IF YP=3 THEN LET YP1=13
3180 GOSUB DRAWP
3190 LET A(XP,YP)=1
3200 RETURN
4999 REM
5000 REM   WINNER?
5001 REM
5005 FOR V=1 TO 2
5010 IF (A(1,1)=V AND A(2,1)=V A
ND A(3,1)=V) OR (A(1,2)=V AND A(
2,2)=V AND A(3,2)=V) OR (A(1,3)=
V AND A(2,3)=V AND A(3,3)=V) OR
(A(1,1)=V AND A(1,2)=V AND A(1,3
)=V) OR (A(2,1)=V AND A(2,2)=V A
ND A(2,3)=V) OR (A(3,1)=V AND A(
3,2)=V AND A(3,3)=V) OR (A(1,1)=
V AND A(2,2)=V AND A(3,3)=V) OR
(A(3,1)=V AND A(2,2)=V AND A(1,3
)=V) THEN GOTO 5100+(100*V)
5020 NEXT V
5022 IF GOES<>9 THEN RETURN
5025 FOR Z=1 TO 100
```

```
5030 PRINT AT 19,2;"IT IS A DRAW
.WE BOTH GET";AT 20,9;"1 POINT."
5040 PRINT AT 19,2;"IT IS A DRAW
.WE BOTH GET";AT 20,9;"1 POINT."
5050 NEXT Z
5055 LET COMPW=COMPW+1
5056 LET HUMAN=HUMAN+1
5060 GOSUB CLEAR
5070 PRINT AT 18,2;"PRESS: C TO
CONTINUE,";AT 20,7;"SPACE BAR TO
 STOP."
5080 IF INKEY$<>"C" THEN GOTO 50
80
5090 CLS
5100 FOR O=1 TO 3
5101 FOR G=1 TO 3
5102 LET A(G,O)=0
5103 NEXT G
5104 NEXT O
5110 GOTO 20
5200 LET HUMAN=HUMAN+2
5205 FOR Z=1 TO 100
5210 PRINT AT 19,6;"YOU WIN 2 PO
INTS.";AT 19,6;"YOU WIN 2 POINTS
."
5215 NEXT Z
5220 GOSUB CLEAR
5230 GOTO 5070
5300 LET COMPW=COMPW+5
5310 FOR Z=1 TO 100
5320 PRINT AT 19,6;"I WIN 5 POIN
TS.";AT 19,6;"I WIN 5 POINTS."
5330 NEXT Z
5340 GOSUB CLEAR
5350 GOTO 5070
5999 REM
6000 REM   ZX-81 MOVES
6001 REM
6005 FOR B=1 TO 11
6006 LET WHERE=WHERE+RND
6007 NEXT B
6008 LET WHERE=WHERE/11
6010 IF WHERE<=.5 THEN GOTO 7000
6020 FOR T=1 TO 3
6030 FOR U=1 TO 3
6040 IF A(U,T)=0 THEN GOTO 6080
6050 NEXT U
6060 NEXT T
6065 GOTO 7000
6070 RETURN
6080 IF U=1 THEN LET XC1=14
6090 IF U=2 THEN LET XC1=19
6100 IF U=3 THEN LET XC1=24
6110 IF T=1 THEN LET YC1=3
6120 IF T=2 THEN LET YC1=8
6130 IF T=3 THEN LET YC1=13
6135 IF A(U,T)<>0 THEN GOTO 6000
6140 LET A(U,T)=2
6150 GOSUB DRAWC
6160 RETURN
7000 FOR T=1 TO 3
7001 FOR U=1 TO 3
7005 LET XC=INT (RND*3)+1
7010 LET YC=INT (RND*3)+1
7020 IF XC=0 OR XC>3 OR YC=0 OR
YC>3 THEN GOTO 7000
7030 IF A(XC,YC)=0 THEN GOTO 608
0
7039 NEXT U
7040 NEXT T
7050 GOTO 6020
7999 REM
8000 REM   GRID
8001 REM
8002 PRINT AT 0,0;
8010 PRINT "
        "
8020 PRINT "  ZX-81
        "
8030 PRINT "
        "
8040 PRINT "  00000
        "
8050 PRINT "
        "
8060 PRINT "   YOU
        "
8070 PRINT "
        "
8080 PRINT "  00000
        "
8090 PRINT "
        "
8100 PRINT "-------------
        "
8110 PRINT "
        "
8120 PRINT "
        "
8130 PRINT "
        "
8140 PRINT "
        "
8150 PRINT "
        "
8160 PRINT "
        "
8170 PRINT "
        "
8200 LET H1=7-(LEN (STR$ COMPW))
8210 LET H2=7-(LEN (STR$ HUMAN))
8220 PRINT AT 3,H1;COMPW;AT 7,H2
;HUMAN
8240 RETURN
8999 REM
9000 REM   INITIALISE
9001 REM
9010 DIM A(3,3)
9020 FOR T=1 TO 3
9030 FOR Z=1 TO 3
9040 LET A(Z,T)=0
9050 NEXT Z
9060 NEXT T
9070 LET DRAWC=1500
9080 LET DRAWP=1000
9090 LET WIN=5000
9100 LET COMP=6000
9110 LET PLAYER=3000
9120 LET CLEAR=2000
9130 LET GRID=8000
9140 LET HUMAN=0
9150 LET COMPW=0
9160 LET GOES=0
9170 LET WHERE=0
9210 RETURN
```



```
          NOUGHTS

            AND

          CROSSES

          THIS IS YOUR PIECE

          THIS IS THE ZX-81 PIECE

**** PRESS NEWLINE TO START ****

  (C) S.P.STRATFORD, 28/11/82

   THE INTRODUCTION PAGE.
```



```
--------------------------------
  ZX-81
  00000
  HUMAN
  00000
----------
DO YOU
WISH TO
GO FIRST?
 Y OR N.
```

**** THE BOARD ****

David Wilkins

PEEPER

Someone's watching you! A paranoid little program, this.

In this game for your 16KZX81, you are looking up at a four storey building. On each floor, there are five windows... and someone keeps looking out of them at you.

From your vantage point, you have a good view of the windows and whenever a face appears at one of the windows, you have the opportunity to fire a shot at it before it disappears.

GOING STARE CRAZY

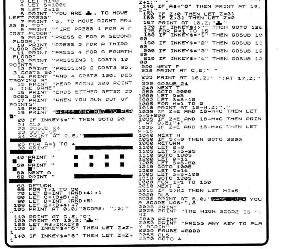
A total of thirty heads could look at you, although the window they will choose will be random each time. When a face

appears at a window, you must position your character directly underneath using the '5' key to go left and the '8' key to go right. Then you can fire at the face; to fire a shot at a head on the first floor, press the '1' key, for a shot at the second floor press the '2' key, and so on.

Each time you hit a head, you score a total of 200 points. However, firing at higher level floors uses up more energy than firing at the lower floors; a first floor shot will cost you 10 points, a second floor shot will count for 25 points, 50 points for a third floor shot and 100 points for a shot at a head on the fourth floor.

You start off the game with 1,000 points, but you'll soon find that a few missed shots at the fourth floor will quickly deplete your score. The game ends after 30 heads have appeared at the window or if you run out of points.

The game also includes a high score feature and instructions for the game's operation are included within the program.



```
   1 REM ***PEEPER***
   3 LET HI=0
   4 LET S=1000
   5 LET Z=15
   6 PRINT "YOU ARE ┻. TO MOVE
LEFT PRESS"
   7 PRINT "5, TO MOVE RIGHT PRE
SS 8. TO"
   8 PRINT "FIRE PRESS 1 FOR A F
IRST FLOOR"
   9 PRINT "PRESS 2 FOR A SECOND
 FLOOR."
  10 PRINT "PRESS 3 FOR A THIRD
FLOOR AND"
  11 PRINT "PRESS 4 FOR A FOURTH
 FLOOR"
  12 PRINT "PRESSING 1 COSTS 10
POINTS, "
  13 PRINT "PRESSING 2 COSTS 25,
 3 COSTS 50"
  14 PRINT "AND 4 COSTS 100. DES
TROYING A"
  15 PRINT "HEAD EARNS 200 POINT
S. THE GAME"
  16 PRINT "ENDS EITHER AFTER 30
 GOES OR"
  17 PRINT "WHEN YOU RUN OUT OF
POINTS"
  18 PRINT
  19 PRINT "PRESS ANY KEY TO PLA
Y"
  20 IF INKEY$="" THEN GOTO 20
  21 CLS
  22 GOSUB 24
  23 GOTO 65
  24 PRINT AT 2,5;"██████████████
██████"
  25 FOR A=1 TO 4
  30 PRINT "     ███████████████
██████"
  40 PRINT "     █   █   █   █
█  █"
  50 PRINT "     █   █   █   █
█  █"
  60 NEXT A
  62 PRINT "     ███████████████
██████"
  63 RETURN
  65 FOR T=1 TO 30
  70 LET B=INT (RND*4)+1
  80 LET C=(3*B)+2
  90 LET D=INT (RND*5)
 100 LET E=(D*4)+7
 105 PRINT AT 0,0;"SCORE: ";S;"
"
 110 PRINT AT C,E;"O"
 120 PRINT AT 18,Z;"┻"
 122 LET A$=INKEY$
 130 IF INKEY$="5" THEN LET Z=Z-
1
 140 IF INKEY$="8" THEN LET Z=Z+
1
 143 IF A$="5" THEN PRINT AT 18,
Z+1;" "
 146 IF A$="8" THEN PRINT AT 18,
Z-1;" "
 150 IF Z<0 THEN LET Z=31
 160 IF Z>31 THEN LET Z=0
 167 PRINT AT 18,Z;"┻"
 170 IF INKEY$<>"" THEN GOTO 120
 175 FOR P=1 TO 15
 180 IF INKEY$="1" THEN GOSUB 10
00
 190 IF INKEY$="2" THEN GOSUB 11
00
 200 IF INKEY$="3" THEN GOSUB 12
00
 210 IF INKEY$="4" THEN GOSUB 13
00
 220 NEXT P
 230 PRINT AT C,E;" "
 233 PRINT AT 16,Z;" ";AT 17,Z;"
"
 235 GOSUB 24
 240 NEXT T
 250 GOTO 2000
1000 LET Q=5
1002 LET S=S-10
1005 FOR H=1 TO Q
1010 PRINT AT 18-H,Z;"."
1030 IF Z=E AND 18-H=C THEN LET
S=S+200
1035 IF Z=E AND 18-H=C THEN PRIN
T AT C,E;"*"
1036 IF Z=E AND 18-H=C THEN LET
P=15
1040 NEXT H
1050 IF S<=0 THEN GOTO 2000
1060 RETURN
1100 LET Q=8
1105 LET S=S-25
1110 GOTO 1005
1200 LET Q=11
1205 LET S=S-50
1210 GOTO 1005
1300 LET Q=14
1305 LET S=S-100
1310 GOTO 1205
2000 FOR I=1 TO 150
2010 NEXT I
2015 IF S>HI THEN LET HI=S
2020 CLS
2030 PRINT AT 6,0;"GAME OVER YOU
R SCORE WAS ";S
2033 PRINT
2035 PRINT "THE HIGH SCORE IS ";
HI
2040 PRINT
2050 PRINT "PRESS ANY KEY TO PLA
Y AGAIN"
2060 PAUSE 40000
2065 CLS
2070 GOTO 4
```

Kenneth Law

FIVE CARD TRICK

You've got to hand it to us — another great game for the ZX81!

This program is a version of the card game, pontoon, for you to type in on your ZX81.

Including all the main features of pontoon, the program starts you off with £50 to gamble and the game will continue until you run out of cash.

TWISTING THE NIGHT AWAY

As in the card game you will be dealt a single card on which you must decide how much you would like to place a bet. Your second card will then be dealt to you. Should you be dealt two aces, the program automatically assumes the first ace is worth one and asks you whether you would like the second ace to be worth one or 11.

With two cards in your hand, you must decide whether you want to 'stick' with your hand or 'twist'. If you 'stick', it means you are happy with your hand as it stands and do not require another card. Should you not have enough points in your hand, you can 'twist', which means you are dealt another card which will get you nearer the magic number of 21. Should your cards total more than 21, the computer will tell you that you have 'busted'.

Once you have stuck with your hand, the computer will work out its own hand and you will be told whether you have won or not. In the event of you and the computer both having hands which total the same, the computer will win — it's known in the 'trade' as 'banker's advantage'.

When you decide you have had enough of beating the computer at its own game, or you've been absolutely thrashed and you've lost all your money, you will be told your ranking as a pontoon player.

```
   1 REM PONTOONS
BY    KENNETH M. LAW,ALEXANDRIA
   2 RAND
   3 GOSUB 8000
   4 LET M=50
   5 CLS
   6 LET Y$="10"
   7 LET A$="N"
   8 LET O$="N"
   9 LET Z$="N"
  10 LET A=0
  20 LET B=0
  30 LET C=-4
  40 LET E=0
  50 LET F=0
  60 LET G=0
  70 LET R=0
  75 LET U=0
  80 LET S=-4
  85 LET V=0
  87 LET XL=0
  88 LET XB=0
  90 GOTO 170
 100 FOR T=1 TO 3
 110 PRINT AT 20,0;"DO YOU WISH
A TWIST ? (Y OR N)"
 120 LET B$=INKEY$
 130 IF B$="Y" THEN GOTO 350
 140 IF B$="N" THEN GOTO 420
 150 GOTO 120
 160 NEXT T
 170 PRINT "PONTOONS","YOU HAVE
£";M
 180 PRINT AT 3,0;"BANKER""S CAR
DS"
 190 FOR N=1 TO 2
 200 PRINT "████ ████ ████ █
█ ██"
 210 PRINT "█▚▚█ █▚▚█ █▚▚█ █
▚█ █▚▚█"
 220 PRINT "█▘?▟█ █▘?▟█ █▘?▟█ █
?▟█ █▘?▟█"
 230 PRINT "█▚▚█ █▚▚█ █▚▚█ █
▚█ █▚▚█"
 240 PRINT "████ ████ ████ █
██ ██"
 250 PRINT AT 13,0;"YOUR CARDS"
 260 NEXT N
 270 FOR W=1 TO 2
 275 LET P=1
 280 LET B=INT (RND*13)+1
 282 IF B=1 THEN GOTO 1000
 284 IF B>=10 THEN GOTO 1500
 290 LET A=A+B
 295 LET U=U+B
 300 LET C=C+6
 310 PRINT AT 16,C;B
 315 IF C=2 THEN GOTO 900
 320 NEXT W
 330 IF A=21 OR U=21 THEN LET A$
="O"
 340 IF A<=21 OR U<=21 THEN GOTO
100
 350 FOR K=1 TO 3
 351 LET P=2
 353 RAND
 357 LET B=INT (RND*13)+1
 358 IF B=1 THEN GOTO 1000
 359 IF B>=10 THEN GOTO 1500
 360 LET A=A+B
 365 LET U=U+B
 370 LET C=C+6
 380 PRINT AT 16,C;B
 390 IF C=26 THEN LET G=5
 400 IF A>21 AND O$<>"Y" THEN GO
TO 800
 403 IF A>21 AND U>21 AND O$="Y"
THEN GOTO 800
 405 IF C=26 THEN GOTO 420
 410 IF A<=21 OR U<=21 THEN GOTO
160
 415 NEXT K
 420 FOR Z=1 TO 2
 423 LET OO=1
 425 PRINT AT 20,0;"
                         "
 427 RAND
 430 LET D=INT (RND*13)+1
 433 IF D=1 THEN GOTO 3000
 436 IF D>=10 THEN GOTO 3500
 440 LET R=R+D
 445 LET V=V+D
 450 LET S=S+6
 460 PRINT AT 6,S;D
 470 NEXT Z
 480 IF R=21 OR V=21 THEN GOTO 6
80
 490 IF A$="O" THEN GOTO 700
 500 IF R>21 AND Z$<>"Y" THEN GO
TO 820
 503 IF R>21 AND V>21 AND Z$="Y"
THEN GOTO 820
 510 IF R>=16 AND R<=21 THEN GOT
O 620
 515 IF V>=16 AND V<=21 THEN GOT
O 620
 520 FOR Q=1 TO 3
 523 LET OO=2
 525 RAND
 530 LET D=INT (RND*13)+1
 533 IF D=1 THEN GOTO 3000
 536 IF D>=10 THEN GOTO 3500
 540 LET R=R+D
 545 LET V=V+D
 550 LET S=S+6
 560 PRINT AT 6,S;D
 570 IF S=26 THEN LET E=5
 575 GOTO 581
 580 IF R>21 AND Z$<>"Y" THEN GO
TO 820
 581 IF R>21 AND V>21 AND Z$="Y"
THEN GOTO 820
 583 IF R>21 THEN LET R=V
 584 IF V>21 THEN LET V=R
 585 IF S=26 THEN GOTO 620
 586 IF U<=21 AND U>A THEN LET A
=U
 587 IF A>21 THEN LET A=U
 589 LET H=INT (RND*2)
 590 IF H=1 AND A>=16 AND A<=21
THEN GOTO 620
 595 IF H=1 AND U>=16 AND U<=21
THEN GOTO 620
 596 IF A<16 AND U<16 THEN GOTO
610
 597 IF H=2 AND A<20 AND U<20 TH
EN GOTO 610
 600 GOTO 620
 610 NEXT Q
 620 IF U<=21 AND U>A THEN LET A
=U
 622 IF A>21 THEN LET A=U
 626 IF V<=21 AND V>R THEN LET R
=V
 628 IF R>21 THEN LET R=V
 630 IF E=5 AND G<>5 THEN GOTO 7
20
 635 IF G=5 AND E<>5 THEN GOTO 7
40
 640 IF E=5 AND G=5 AND R>=A THE
N GOTO 720
 650 IF E=5 AND G=5 AND A>R THEN
 GOTO 740
 660 IF R>=A THEN GOTO 760
 670 IF A>R THEN GOTO 780
 680 PRINT AT 10,0;"PONTOONS.BAN
KER WINS."
 685 LET M=M-X
 690 GOTO 840
 700 PRINT AT 10,0;"PONTOONS.YOU
 WIN."
 705 LET M=M+X
 710 GOTO 840
 720 PRINT AT 10,0;"BANKER WINS
WITH                  A FIVE CARD
TRICK OF ";R;"."
 725 LET M=M-X
 730 GOTO 840
 740 PRINT AT 10,0;"YOU WIN WITH
                      A FIVE CARD
TRICK OF ";A;"."
 745 LET M=M+X
 750 GOTO 840
 760 PRINT AT 10,0;"BANKER WINS
WITH ";R;"."
 765 LET M=M-X
 770 GOTO 840
 780 PRINT AT 10,0;"YOU WIN WITH
 ";A;"."
 785 LET M=M+X
 790 GOTO 840
 800 PRINT AT 10,0;"BURST.BANKER
 WINS."
 805 LET M=M-X
 810 GOTO 840
 820 PRINT AT 10,0;"BURST.YOU WI
N."
 825 LET M=M+X
 840 PRINT AT 20,0;"DO YOU WISH
ANOTHER GAME?(Y/N)"
 850 LET B$=INKEY$
 860 IF B$="Y" THEN GOTO 5
 870 IF B$="N" THEN GOTO 9000
 880 GOTO 850
 900 PRINT AT 20,0;"HOW MUCH DO
YOU WANT TO GAMBLE ?"
 910 IF M<=0 THEN PRINT AT 20,0;
"YOU HAVE NO MORE MONEY TO GAMBL
E WITH."
 915 IF M<=0 THEN PAUSE 50
 920 IF M<=0 THEN GOTO 9000
 930 INPUT X
 940 IF X>M OR X<=0 THEN GOTO 93
0
 950 PRINT AT 1,16;"THIS GAME IS
"
 960 PRINT AT 2,18;"FOR £";X
 970 GOTO 320
1000 IF XL=1 THEN GOTO 2000
1010 LET XL=1
1020 LET O$="Y"
1030 LET A=A+B
1040 IF B=1 THEN LET U=U+11
1050 IF B=11 THEN LET U=U+1
1060 LET C=C+6
1070 PRINT AT 16,C;"A"
1080 IF P=1 THEN GOTO 315
1090 IF P=2 THEN GOTO 390
1500 LET A=A+10
1510 LET U=U+10
1520 LET C=C+6
1530 LET Y=INT (RND*4)+1
1540 IF Y=1 THEN LET Y$="10"
1550 IF Y=2 THEN LET Y$="J"
1560 IF Y=3 THEN LET Y$="Q"
1570 IF Y=4 THEN LET Y$="K"
1580 PRINT AT 16,C;Y$
1590 IF P=1 THEN GOTO 315
1600 IF P=2 THEN GOTO 390
2000 LET A=A+1
2010 LET U=U+1
2020 LET C=C+6
2030 PRINT AT 16,C;"A"
2040 IF P=1 THEN GOTO 315
2050 IF P=2 THEN GOTO 390
3000 IF XB=1 THEN GOTO 4000
3010 LET XB=1
3020 LET Z$="Y"
3030 LET R=R+D
3040 IF D=1 THEN LET V=V+11
3050 IF D=11 THEN LET V=V+1
3060 LET S=S+6
3070 PRINT AT 6,S;"A"
3080 IF QQ=1 THEN GOTO 470
3090 IF QQ=2 THEN GOTO 570
3500 LET R=R+10
3510 LET V=V+10
3520 LET S=S+6
3530 LET Y=INT (RND*4)+1
3540 IF Y=1 THEN LET Y$="10"
3550 IF Y=2 THEN LET Y$="J"
3560 IF Y=3 THEN LET Y$="Q"
3570 IF Y=4 THEN LET Y$="K"
3580 PRINT AT 6,S;Y$
3590 IF QQ=1 THEN GOTO 470
3600 IF QQ=2 THEN GOTO 570
4000 LET R=R+1
4010 LET V=V+1
4020 LET S=S+6
4030 PRINT AT 6,S;"A"
4040 IF QQ=1 THEN GOTO 470
4050 IF QQ=2 THEN GOTO 570
8000 FOR J=1 TO 50
8010 PRINT AT 10,10;"PONTOONS"
8020 PRINT AT 12,6;"BY KENNETH L
AW"
8030 PRINT AT 16,0;"YOU HAVE BEE
N GIVEN £50 TO        GAMBLE WITH,
COURTESY OF KENNETH LAW."
8050 PRINT AT 10,10;"PONTOONS"
8060 PRINT AT 12,6;"BY KENNETH L
AW"
8100 NEXT J
8200 CLS
8300 RETURN
9000 CLS
9010 PRINT AT 5,0;"YOU CAME OUT
OF THE CASINO WITH £";M
9020 IF M>50 THEN PRINT ,,,,"YOU
 MADE A PROFIT OF £";(M-50)
9030 IF M=50 THEN PRINT ,,,,"YOU
 DID NOT LOOSE ANYTHING."
9040 IF M<50 THEN PRINT ,,,,"YOU
 LOST £";(50-M)
9050 PRINT ,,,,"*****************
*********************** PONTOON""S
 RATING ******************************
******************"
9060 IF M>500 THEN PRINT ,,,,"YO
U ARE AN EXPERT,A PROFESSIONAL A
T WORK."
9065 IF M>500 THEN GOTO 9200
9070 IF M>150 THEN PRINT ,,,,"YO
U ARE A GOOD CONSISTANCY      PL
AYER WITH WORKING GREY MATTER."
9075 IF M>150 THEN GOTO 9200
9080 IF M>50 THEN PRINT ,,,,"NOT
 BAD,AT LEAST YOU MANAGED TO BRE
AK EVENS."
9085 IF M>50 THEN GOTO 9200
9090 IF M=50 THEN PRINT ,,,,"YOU
 ARE NOT A GAMBLER,YOU ARE A MIS
ER."
9095 IF M=50 THEN GOTO 9200
9100 IF M<50 THEN PRINT ,,,,"ABS
OLUTELY PATHETIC-THAT""S YOU, MA
TE-MAY I SUGGEST A NEW HAT    MA
YBE ONE WHICH HAS A BIG ■ ON  IT
."
9200 STOP
```

PROFIT PROPHET

A serious program to help you get to grips with your business.

The object of this program is to help a company or financial institution measure their profitability.

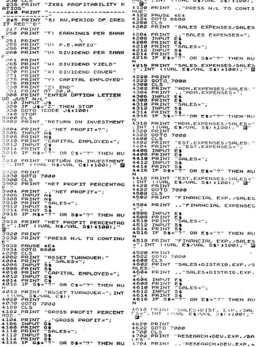
On RUNning the program you will be greeted with a menu which will provide a route through to the particular ratio you require. Here follows a list of the facilities available with this program:

a) Return on investment.
b) Net profit percentage.
c) Asset turnover.
d) Gross profit percentage.
e) Selling expenses/sales.
f) Administration expenses/sales.
g) Establishment expenses/sales.
h) Financial expenses/sales.
i) Sales and distribution expenses/sales.
j) Research and development expenses/sales.
k) Fixed asset turnover.
l) Current asset turnover.
m) Working capital turnover.
n) Debtor's turnover.
o) Rate of stock turnover.
p) Current asset ratio.
q) Acid test.
r) Average period of credit given.
s) Average period of credit received.
t) Earnings per share.
u) P.E. ratio.
v) Dividend per share.
w) Dividend yield.
x) Dividend cover.
y) Capital employed.

As you can see from the long list of options, the program certainly tries to fill in the gaps of your business knowledge.

In fact, the author has added a number of program notes which will be revealed should you choose certain options. These are not part of the operation of the program but illustrate certain patterns which developed over a period of trading months so were added for the author's convenience. Should these not apply to you they could be left out or have new comments substituted.

```
  10 GOSUB 9000
  32 PAUSE 90
  34 CLS
  40 PRINT "ZX81 PROFITABILITY M
EASURES"
  45 PRINT "--------------------
------------"
  50 PRINT "A) RETURN ON INVESTM
ENT"
  51 PRINT
  55 PRINT "B) NET PROFIT PERCEN
TAGE"
  56 PRINT
  60 PRINT "C) ASSET TURNOVER"
  61 PRINT
  65 PRINT "D) GROSS PROFIT PERC
ENTAGE"
  66 PRINT
  70 PRINT "E) SELLING EXPENSES/
SALES"
  71 PRINT
  75 PRINT "F) ADMINISTRATION EX
PENSES/SALES"
  76 PRINT
  80 PRINT "G) ESTABLISHMENT EXP
ENSES/SALES"
  81 PRINT
  85 PRINT "H) FINANCIAL EXPENSE
S/SALES"
  86 PRINT
  90 PRINT "I) SALES+DISTRIB.EXP
/SALES"
  91 PRINT
 100 PRINT "ENTER OPTION LETTER
OR JUST N/L"
 110 INPUT J$
 120 IF J$="" THEN GOTO 130
 125 GOTO (CODE J$*100)
 130 CLS
 135 PRINT "ZX81 PROFITABILITY R
ATIOS"
 140 PRINT "--------------------
------------"
 150 PRINT "J) RES.+DEVELOPMENT
EXP. SALES"
 151 PRINT
 155 PRINT "K) FIXED ASSET TURNO
VER"
 156 PRINT
 160 PRINT "L) CURRENT ASSET TUR
NOVER"
 161 PRINT
 165 PRINT "M) WORKING CAPITAL T
URNOVER"
 166 PRINT
 170 PRINT "N) DEBTORS TURNOVER"
 171 PRINT
 175 PRINT "O) RATE OF STOCK TUR
NOVER"
 176 PRINT
 180 PRINT "P) CURRENT ASSET RAT
IO"
 181 PRINT
 185 PRINT "Q) ACID TEST"
 186 PRINT
 190 PRINT "R) AV.PERIOD OF CRED
IT GIVEN"
 196 PRINT
 200 PRINT "ENTER OPTION LETTER
OR JUST N/L"
```



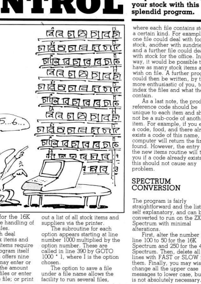
```
 210 INPUT J$
 220 IF J$="" THEN GOTO 230
 225 GOTO (CODE J$+100)
 230 CLS
 235 PRINT "ZX81 PROFITABILITY R
ATIOS"
 240 PRINT "--------------------
-----------"
 245 PRINT "S) AV.PERIOD OF CRED
IT REC""D"
 247 PRINT
 250 PRINT "T) EARNINGS PER SHAR
E"
 251 PRINT
 255 PRINT "U) P.E.RATIO"
 256 PRINT
 260 PRINT "V) DIVIDEND PER SHAR
E"
 261 PRINT
 265 PRINT "W) DIVIDEND YIELD"
 266 PRINT
 270 PRINT "X) DIVIDEND COVER"
 271 PRINT
 275 PRINT "Y) CAPITAL EMPLOYED"
 276 PRINT
 280 PRINT "Z) END"
 281 PRINT AT 20,0;
 300 PRINT "ENTER OPTION LETTER
OR JUST N/L"
 310 INPUT J$
 320 IF J$="Z" THEN STOP
 330 GOTO (CODE J$+100)
 340 STOP
3800 CLS
3802 PRINT "RETURN ON INVESTMENT
"
3804 PRINT ,,"NET PROFIT=?";
3806 INPUT N$
3808 PRINT N$
3810 PRINT "CAPITAL EMPLOYED=";
3812 INPUT C$
3814 PRINT C$
3816 IF N$="?" OR C$="?" THEN RU
N
3818 PRINT "RETURN ON INVESTMENT
=";INT ((VAL N$/VAL C$)*100);" ■
"
3820 PRINT
3830 GOTO 7000
3900 CLS
3902 PRINT "NET PROFIT PERCENTAG
E:"
3904 PRINT ,,"NET PROFIT=";
3906 INPUT N$
3908 PRINT N$
3910 PRINT "SALES=";
3912 INPUT S$
3914 PRINT S$
3916 IF N$="?" OR S$="?" THEN RU
N
3918 PRINT "NET PROFIT PERCENTAG
E=";INT ((VAL N$/VAL S$)*100);"
■"
3920 PRINT
3930 PRINT "PRESS N/L TO CONTINU
E"
3932 PAUSE 4E4
3934 GOTO 6600
4000 CLS
4002 PRINT "ASSET TURNOVER:"
4004 PRINT ,,"SALES=";
4006 INPUT S$
4008 PRINT S$
4010 PRINT "CAPITAL EMPLOYED=";
4012 INPUT C$
4014 PRINT C$
4016 IF S$="?" OR C$="?" THEN RU
N
4018 PRINT "ASSET TURNOVER=";INT
 ((VAL S$/VAL C$))
4020 PRINT
4030 GOTO 7000
4100 CLS
4102 PRINT "GROSS PROFIT PERCENT
AGE:"
4104 PRINT ,,"GROSS PROFIT=";
4106 INPUT G$
4108 PRINT G$
4110 PRINT "SALES=";
4112 INPUT S$
4114 PRINT S$
4116 IF G$="?" OR S$="?" THEN RU
N
4118 PRINT "GROSS PROFIT PERCENT
AGE=";INT ((VAL G$/VAL S$)*100);
" ■"
4120 PRINT ,,"PRESS N/L TO CONTI
NUE"
4122 PAUSE 4E4
4124 GOTO 6600
4200 CLS
4202 PRINT "SALES EXPENSES/SALES
:"
4204 PRINT ,,"SALES EXPENSES=";
4206 INPUT E$
4208 PRINT E$
4210 PRINT "SALES=";
4212 INPUT S$
4214 PRINT S$
4216 IF E$="?" OR S$="?" THEN RU
N
4218 PRINT "SALES EXPENSES/SALES
=";INT ((VAL E$/VAL S$)*100);" ■
"
4220 PRINT
4222 GOTO 7000
4300 CLS
4302 PRINT "ADM.EXPENSES/SALES:"
4304 PRINT ,,"ADM.EXPENSES=";
4306 INPUT E$
4308 PRINT E$
4310 PRINT "SALES=";
4312 INPUT S$
4314 PRINT S$
4316 IF S$="?" OR E$="?" THEN RU
N
4318 PRINT "ADM.EXPENSES/SALES="
;INT ((VAL E$/VAL S$)*100);" ■"
4320 PRINT
4322 GOTO 7000
4400 CLS
4402 PRINT "EST.EXPENSES/SALES:"
4404 PRINT ,,"EST.EXPENSES=";
4406 INPUT E$
4408 PRINT E$
4410 PRINT "SALES=";
4412 INPUT S$
4414 PRINT S$
4416 IF S$="?" OR E$="?" THEN RU
N
4418 PRINT "EST.EXPENSES/SALES="
;INT ((VAL E$/VAL S$)*100);" ■"
4420 PRINT
4422 GOTO 7000
4500 CLS
4502 PRINT "FINANCIAL EXP./SALES
:"
4504 PRINT ,,"FINANCIAL EXPENSES
=";
4506 INPUT E$
4508 PRINT E$
4510 PRINT "SALES=";
4512 INPUT S$
4514 PRINT S$
4516 IF S$="?" OR E$="?" THEN RU
N
4518 PRINT "FINANCIAL EXP./SALES
=";INT ((VAL E$/VAL S$)*100);" ■
"
4520 PRINT
4522 GOTO 7000
4600 CLS
4602 PRINT "SALES+DISTRIB.EXP./S
ALES:"
4604 PRINT ,,"SALES+DISTRIB.EXP.
=";
4606 INPUT E$
4608 PRINT E$
4610 PRINT "SALES=";
4612 INPUT S$
4614 PRINT S$
4616 IF S$="?" OR E$="?" THEN RU
N
4618 PRINT "SALES+DIST. EXP./SAL
ES=";INT ((VAL E$/VAL S$)*100);"
■"
4620 PRINT
4622 GOTO 7000
4700 CLS
4702 PRINT "RESEARCH+DEV.EXP./SA
LES:"
4704 PRINT ,,"RESEARCH+DEV.EXP.=
4706 INPUT E$
4708 PRINT E$
4710 PRINT "SALES=";
4712 INPUT S$
4714 PRINT S$
4716 IF E$="?" OR S$="?" THEN RU
N
4718 PRINT "RESEARCH+DEV. EXP./S
ALES="; INT ((VAL E$/VAL S$)*100)
," █"
4720 PRINT
4722 GOTO 7000
4800 CLS
4802 PRINT "FIXED ASSET TURNOVER
"
4804 PRINT ,,"SALES=";
4806 INPUT S$
4808 PRINT S$
4810 PRINT "FIXED ASSETS=";
4812 INPUT F$
4814 PRINT F$
4816 IF S$="?" OR F$="?" THEN RU
N
4818 PRINT "FIXED ASSET TURNOVER
=";INT ((VAL S$/VAL F$))
4820 PRINT ,,"PRESS N/L TO CONTI
NUE"
4822 PAUSE 4E4
4824 GOTO 6500
4900 CLS
4902 PRINT "CURRENT ASSET TURNOV
ER:"
4904 PRINT ,,"SALES=";
4906 INPUT S$
4908 PRINT S$
4910 PRINT "CURRENT ASSETS=";
4912 INPUT C$
4914 PRINT C$
4916 IF S$="?" OR C$="?" THEN RU
N
4918 PRINT "CURRENT ASSET TURNOV
ER=";INT ((VAL S$/VAL C$))
4920 PRINT ,,"PRESS N/L TO CONTI
NUE"
4922 PAUSE 4E4
4924 GOTO 6400
5000 CLS
5002 PRINT "WORKING CAPITAL TURN
OVER:"
5004 PRINT ,,"SALES=";
5006 INPUT S$
5008 PRINT S$
5010 PRINT "WORKING CAPITAL=";
5012 INPUT W$
5014 PRINT W$
5016 IF W$="?" OR S$="?" THEN RU
N
5018 PRINT "WORKING CAPITAL TURN
OVER=";INT ((VAL S$/VAL W$))
5020 PRINT
5022 GOTO 7000
5100 CLS
5102 PRINT "DEBTORS TURNOVER:"
5104 PRINT ,,"CREDIT SALES=";
5106 INPUT S$
5108 PRINT S$
5110 PRINT "TRADE DEBTORS=";
5112 INPUT D$
5114 PRINT D$
5116 IF S$="?" OR D$="?" THEN RU
N
5118 PRINT "DEBTORS TURNOVER=";I
NT ((VAL S$/VAL D$))
5120 PRINT
5122 GOTO 7000
5200 CLS
5202 PRINT "RATE OF STOCK TURNOV
ER:"
5204 PRINT ,,"COST OF SALES=";
5206 INPUT C$
5208 PRINT C$
5209 IF C$="?" THEN RUN
5210 PRINT "OPENING STOCK=";
5212 INPUT S1
5214 PRINT S1
5216 PRINT "CLOSING STOCK=";
5218 INPUT S2
5220 PRINT S2
5222 IF S1=0 THEN LET S$=STR$ S2
5224 IF S2=0 THEN LET S$=STR$ S1
5226 IF S1<>0 AND S2<>0 THEN LET
 S$=STR$ (0.5*(S1+S2))
5228 PRINT "RATE OF STOCK TURNOV
ER=";
5230 LET X=INT (VAL C$/VAL S$)
5232 PRINT INT (365/X);"DAYS"
5234 PRINT
5236 GOTO 7000
5300 CLS
5302 PRINT "CURRENT ASSET RATIO
:"
5304 PRINT ,,"CURRENT ASSETS=";
5306 INPUT C$
5307 PRINT C$
5308 PRINT "CURRENT LIABILITIES=
";
5310 INPUT L$
5312 PRINT L$
5314 IF C$="?" OR L$="?" THEN RU
N
5316 PRINT "CURRENT ASSET RATIO=
";INT ((VAL C$/VAL L$));":1"
5318 PRINT
5320 GOTO 7000
5400 CLS
5402 PRINT "ACID TEST:"
5404 PRINT ,,"CURRENT ASSETS=";
5406 INPUT F$
5408 PRINT F$
5410 PRINT "STOCK=";
5412 INPUT S$
5414 PRINT S$
5416 PRINT "CURRENT LIABILITIES=
";
5418 INPUT L$
5420 PRINT L$
5422 PRINT "ACID TEST=";INT ((VA
L F$-VAL S$)/VAL L$);":1"
5424 PRINT ,,"PRESS N/L TO CONTI
NUE"
5426 PAUSE 4E4
5428 GOTO 6300
5500 CLS
5502 PRINT "AVERAGE PERIOD OF CR
EDIT GIVEN:"
5504 PRINT ,,"DEBTORS=";
5506 INPUT D$
5508 PRINT D$
5510 PRINT "CREDIT SALES=";
5512 INPUT C$
5514 PRINT C$
5516 IF C$="?" OR D$="?" THEN RU
N
5518 PRINT "AV.PERIOD OF CREDIT
GIVEN="
5520 PRINT "          ";INT ((VAL
 D$/VAL C$)*365);" DAYS"
5530 PRINT
5540 GOTO 7000
5600 CLS
5602 PRINT "AVERAGE PERIOD OF CR
EDIT REC""D:"
5604 PRINT ,,"CREDITORS=";
5606 INPUT C$
5608 PRINT C$
5610 PRINT "PURCHASES=";
5612 INPUT P$
5614 PRINT P$
5616 IF C$="?" OR P$="?" THEN RU
N
5618 PRINT "AV.PERIOD OF CREDIT
REC""D="
5620 PRINT "          ";INT ((VAL
 C$/VAL P$)*365);" DAYS"
5630 PRINT
5640 GOTO 7000
5700 CLS
5702 PRINT "EARNINGS PER SHARE:"
5704 PRINT ,,"NET PROFIT AFTER T
AX AND FIXED  INTREST=";
5706 INPUT N$
5708 PRINT N$
5710 PRINT "NO.OF ORD SHARES ISS
UED=";
5712 INPUT S$
5714 PRINT S$
5716 PRINT "EARNINGS PER SHARE="
;INT ((VAL N$/VAL S$)*100)
5718 PRINT
5720 GOTO 7000
5800 CLS
5802 PRINT "PRICE EARNINGS RATIO
:"
5804 PRINT ,,"MARKET PRICE PER S
```

```
HARE=";
5806 INPUT M$
5808 PRINT M$
5810 PRINT "EARNINGS PER SHARE="
;
5812 INPUT S$
5814 PRINT S$
5816 IF M$="?" OR S$="?" THEN RU
N
5818 PRINT "P.E.RATIO=";INT ((VA
L M$/VAL S$))
5820 PRINT
5830 GOTO 7000
5900 CLS
5902 PRINT "DIVIDEND PER SHARE:"
5904 PRINT ,,"DIVIDEND PAID=";
5906 INPUT D$
5907 PRINT D$
5908 PRINT "NO.OF SHARES=";
5910 INPUT N$
5920 PRINT N$
5925 IF D$="?" OR N$="?" THEN RU
N
5930 PRINT "DIVIDEND PER SHARE="
;INT ((VAL D$/VAL N$))
5932 PRINT
5934 GOTO 7000
6000 CLS
6002 PRINT "DIVIDEND YIELD:"
6004 PRINT ,,"DIVIDEND PER SHARE
=";
6006 INPUT D$
6008 PRINT D$
6010 PRINT "MARKET PRICE PER SHA
RE=";
6012 INPUT M$
6014 PRINT M$
6016 IF D$="?" OR M$="?" THEN RU
N
6018 PRINT "DIVIDEND YIELD=";INT
 ((VAL D$/VAL M$)*100);" %"
6020 PRINT
6022 GOTO 7000
6100 CLS
6102 PRINT "DIVIDEND COVER:"
6104 PRINT ,,"NET PROFIT - TAX -
 FIXED INTREST  - PREF.DIVIDEND=
";
6106 INPUT D$
6108 PRINT D$
6110 PRINT "DIV.PAID ON ORD.SHAR
ES=";
6112 INPUT M$
6114 PRINT M$
6116 PRINT "DIVIDEND COVER=";INT
 ((VAL D$/VAL M$))
6118 PRINT
6120 GOTO 7000
6200 CLS
6202 PRINT "CAPITAL EMPLOYED:"
6204 PRINT ,,"SHARE CAPITAL=";
6206 INPUT B$
6208 PRINT B$
6210 PRINT "RESERVES=";
6212 INPUT C$
6214 PRINT C$
6216 PRINT "LONG TERM LIABILITIE
S=";
6218 INPUT D$
6220 PRINT D$
6222 PRINT "INTANGIBLE ASSETS=";
6224 INPUT E$
6226 PRINT E$
6228 PRINT "CAP.EMPLOYED=";((VAL
 C$+VAL D$+VAL B$)-VAL E$)
6230 PRINT
6240 GOTO 7000
6300 CLS
6305 PRINT "ACID TEST:"
6310 PRINT "TOO MUCH RESOURCES B
EING HELD INLIQUID FORM AND THER
EFORE NOT  CONTRIBUTING TO FIRM
S PROFITS."
6320 PRINT ,,"EXAMPLES SUCH AS:"
6330 PRINT ,,"    1) CASH AT BANK,
 AND          2) DEBTORS."
5340 PRINT AT 20,0;"PRESS J,N/L
TO STOP OR                   N/L
TO RE-START"
6350 INPUT J$
5360 IF J$="J" THEN STOP
```

```
6370 RUN
6400 CLS
6405 PRINT "CURRENT ASSET TURNOV
ER:"
6410 PRINT "A FALL IN CURRENT AS
SET TURNOVERINDICATES A PROBLEM
IN SALES OR CURRENT ASSETS:"
6420 PRINT ,,"THERE IS ALSO A PO
SSIBILITY OF  ""UNDERTRADING""."
5400 PRINT AT 20,0;"PRESS J,N/L
TO STOP OR                   N/L
TO RE-START"
6450 INPUT J$
6460 IF J$="J" THEN STOP
6470 RUN
6500 CLS
6505 PRINT "FIXED ASSET TURNOVER
:"
6510 PRINT "A FALL IN F.A.TURNOV
ER INDICATESUNDERUTILISATION OF
CAPACITY."
6515 PRINT "  -EG:PLANT MAY BE TO
O BIG IN             RELATION TO VOL
UME OF SALES"
6520 PRINT ,,"     TOO RAPID EXPA
NSION"
6525 PRINT ,,"     ANTICIPATE DE
MAND FOR SALES     IS JUST NOT TH
ERE."
6540 PRINT AT 20,0;"PRESS J,N/L
TO STOP OR                   N/L
TO RE-START"
6550 INPUT J$
6560 IF J$="J" THEN STOP
6570 RUN
6600 CLS
6605 PRINT "NET+GROSS PROFIT RIS
E:"
6606 PRINT
6610 PRINT "INDICATES OVERTRADIN
G AND A      DELAY BETWEEN PAYMEN
T OF DEBTS   AND RECEIPT OF MONEY
 OWED."
6620 PRINT ,,"REMEDIES FOR FALLI
NG             PROFIT PERCENTAGES
:"
6625 PRINT
6630 PRINT "1)  INCREASE SELLING
PRICE         2)  TIGHTEN SECURITY
ON CASH           AND STOCK
              3)  CHECK ON SLOW MOV
ING STOCK"
6635 PRINT "4)  CHECK STOCK VALUA
TION METHODS5)  CHECK STAFF EFFIC
IENCY         6)  FIRE BUYER AND/OR
 SALES            MANAGER"
5640 PRINT AT 20,0;"PRESS J,N/L
TO STOP OR                   N/L
TO RE-START"
6650 INPUT J$
6660 IF J$="J" THEN STOP
6670 RUN
7000 PRINT "PRESS J,N/L TO END"
                    N/L TO RE-ST
ART"
7010 INPUT J$
7020 IF J$="J" THEN STOP
7030 RUN
7040 PRINT END
9000 CLS
9010 PRINT "********************
************"
9020 PRINT "*";TAB 31;"*"
9030 PRINT "*  ZX81 PROFITABILIT
Y RATIOS        *"
9040 PRINT "*";TAB 31;"*"
9050 PRINT "*      BY MICHAEL CAR
ROLL            *"
9060 PRINT "*";TAB 31;"*"
9070 PRINT "*      RUNS ON A ZX81
 IN 16K         *"
9080 PRINT "*";TAB 31;"*"
9090 PRINT "*         USES APPROX
.10K            *"
9100 PRINT "*";TAB 31;"*"
9110 PRINT "********************
************"
9120 PRINT AT 15,0;"PRESS NEWLIN
E TO CONTINUE..."
9130 INPUT A$
9140 RETURN
```

STOCK CONTROL

Keep efficient control of your stock with this splendid program.



This program for the 16K ZX81 allows the handling of stock control files. Each file set up will deal with up to 100 stock items and will tell you which items require reordering. The program itself is menu driven and offers nine main options: you may enter or delete stock; enter the amount re-ordered; delete files or enter new items; save the file; or print out a list of all stock items and suppliers via the printer.

The subroutine for each option appears starting at line number 1000 multiplied by the option number. These are called in line 390 by GOTO 1000 * I, where I is the option chosen.

The option to save a file under a file name allows the facility to run several files, where each file contains stock of a certain kind. For example, one file could deal with food stock, another with sundries, and a further file could deal with stock for the office. In this way, it would be possible to have as many stock items as you wish on file. A further program could then be written, by the more enthusiastic of you, to index the files and what they contain.

As a last note, the product reference code should be unique to each item and should not be a sub-code of another item. For example, if you enter a code, food, and there already exists a code of this name, the computer will return the first found. However, the entry of the new items routine will tell you if a code already exists, so this should not cause any problem.

SPECTRUM CONVERSION

The program is fairly straightforward and the listing self explanatory, and can be converted to run on the ZX Spectrum with minimal alterations.

First, alter the number in line 100 to 50 for the 16K Spectrum and 250 for the 48K Spectrum. Then, delete all the lines with FAST or SLOW in them. Finally, you may wish to change all the upper case messages to lower case, but this is not absolutely necessary.

```
  10 SAVE "STOC█"
  80 LET Z$="ENTER PRODUCT REFER
ENCE CODE."
 100 LET NO=100
 110 DIM A$(NO,32)
 112 DIM S$(NO,32)
 115 DIM A(100,4)
 200 SLOW
 210 PRINT TAB 9;"STOCK CONTROL.
";TAB 9;"▀▀▀▀▀▀ ▀▀▀▀▀▀▀"
 220 PRINT "OPTIONS:-"
 230 PRINT
 240 PRINT TAB 2;"1. ENTER NEW S
TOCK ITEM."
 250 PRINT TAB 2;"2. PRINT ITEM
DETAILS."
 260 PRINT TAB 2;"3. DELETE OLD
STOCK ITEM."
 270 PRINT TAB 2;"4. ENTER REMOV
ED STOCK."
```

```
 280 PRINT TAB 2;"5. ENTER STOCK
ON ORDER."
 290 PRINT TAB 2;"6. ENTER STOCK
RECEIVED."
 300 PRINT TAB 2;"7. PRINT ITEMS
TO RE-ORDER."
 310 PRINT TAB 2;"8. SAVE STOCK
FILE."
 315 PRINT TAB 2;"9. PRINT ALL R
EF. CODES";TAB 5;"AND SUPPLIERS.
"
 320 PRINT ,,"CHOOSE OPTION (1-9
)."
 330 PRINT TAB 9;"▀▀▀▀▀▀▀▀▀▀▀▀▀
"
 340 IF INKEY$<>"" THEN GOTO 340
 350 IF INKEY$="" THEN GOTO 350
 360 LET O$=INKEY$
 370 IF O$<"1" OR O$>"9" THEN GO
TO 340
 380 LET I=VAL O$
 390 GOTO 1000*I
1000 CLS
1010 PRINT Z$
1020 INPUT B$
1025 FAST
1030 FOR I=1 TO NO
1040 IF A$(I, TO LEN B$)=B$ THEN
 GOTO 1080
1050 IF A$(I,1 TO 4)="    " THEN
 GOTO 1100
1055 NEXT I
1065 SLOW
1070 PRINT ,,"NO MEMORY LEFT."
1075 PRINT ,,"PRESS ANY KEY TO R
ETURN TO MENU."
1076 IF INKEY$<>"" THEN GOTO 107
6
1077 IF INKEY$="" THEN GOTO 1077
1078 CLS
1079 GOTO 210
1080 SLOW
1085 PRINT ,,"STOCK ITEM ALLREAD
Y EXISTS."
1090 GOTO 1075
1100 SLOW
1102 PRINT "██ ";B$
1110 LET A$(I)=B$
1120 PRINT ,,"ENTER AMOUNT IN ST
OCK."
1130 INPUT A(I,1)
1140 IF A(I,1)<0 THEN GOTO 1130
1150 PRINT "██ ";A(I,1)
1160 PRINT ,,"ENTER AMOUNT ON OR
DER."
1170 INPUT A(I,2)
1180 IF A(I,2)<0 THEN GOTO 1170
1190 PRINT "██ ";A(I,2)
1200 PRINT ,,"ENTER RE-ORDER LEV
EL."
1210 INPUT A(I,3)
1220 IF A(I,3)<0 THEN GOTO 1210
1230 PRINT "██ ";A(I,3)
1240 PRINT ,,"ENTER RE-ORDER QUA
NTITY."
1250 INPUT A(I,4)
1260 IF A(I,4)<0 THEN GOTO 1250
1270 PRINT "██ ";A(I,4)
1280 PRINT ,,"ENTER SUPPLIERS DE
TAILS."
1290 INPUT S$(I)
1300 IF S$(I, TO 4)="    " THEN
GOTO 1290
1310 PRINT S$(I)
1320 PRINT ,,"ANY FURTHER NEW IT
EMS (Y OR N)?"
1330 IF INKEY$<>"" THEN GOTO 133
0
1340 IF INKEY$="" THEN GOTO 1340
1350 LET Y$=INKEY$
1355 IF Y$="Y" OR Y$="N" THEN CL
S
1360 IF Y$="Y" THEN GOTO 1000
1370 IF Y$="N" THEN GOTO 200
1380 GOTO 1330
2000 CLS
2010 PRINT Z$
2020 INPUT B$
2030 IF B$="" THEN GOTO 2020
2035 FAST
2040 FOR I=1 TO NO
2050 IF A$(I, TO LEN B$)=B$ THEN
 GOTO 2080
2060 NEXT I
2065 SLOW
2070 PRINT ,,"PRODUCT DOES NOT E
XIST ON FILE."
2075 GOTO 1075
2080 CLS
2085 SLOW
2090 PRINT "PRODUCT DETAILS:-"
2100 PRINT ,,A$(I)
2110 PRINT ,,"PHYSICAL STOCK    :
";A(I,1)
2120 PRINT "QUANTITY ON ORDER:";
A(I,2)
2130 PRINT "TOTAL STOCK      :";
A(I,1)+A(I,2)
2140 PRINT "RE-ORDER LEVEL   :";
A(I,3)
2150 PRINT "RE-ORDER QUANTITY:";
A(I,4)
2160 PRINT ,,"SUPPLIER:-"
2170 PRINT S$(I)
2180 PRINT ,,"PRESS ""C"" FOR CO
PY, ANY OTHER  KEY FOR MENU."
2190 IF INKEY$<>"" THEN GOTO 219
0
2200 IF INKEY$="" THEN GOTO 2200
2210 IF INKEY$="C" THEN COPY
2220 CLS
2230 GOTO 200
3000 CLS
3010 PRINT Z$
3020 INPUT B$
3025 FAST
3030 FOR I=1 TO NO
3040 IF A$(I, TO LEN B$)=B$ THEN
 GOTO 3070
3050 NEXT I
3055 SLOW
3060 GOTO 2070
3070 SLOW
3080 PRINT ,,"PRESS ""D"" TO DEL
ETE, ANY OTHER  KEY TO RETURN TO
 THE MENU."
3081 IF INKEY$<>"" THEN GOTO 308
1
3082 IF INKEY$="" THEN GOTO 3082
3083 IF INKEY$="D" THEN GOTO 308
8
3084 CLS
3085 GOTO 200
3088 FAST
3089 FOR J=I+1 TO NO
3090 LET A$(J-1)=A$(J)
3100 LET S$(J-1)=S$(J)
3110 FOR K=1 TO 4
3120 LET A(J-1,K)=A(J,K)
3130 NEXT K
3140 NEXT J
3150 NEXT K
3160 PRINT ,,"PRODUCT DELETED."
3165 SLOW
3170 GOTO 1075
4000 CLS
4010 PRINT Z$
4020 INPUT B$
4030 IF B$="" THEN GOTO 4030
4035 FAST
4040 FOR I=1 TO NO
4050 IF A$(I, TO LEN B$)=B$ THEN
 GOTO 4090
4060 NEXT I
4065 SLOW
4070 GOTO 2070
4090 SLOW
4091 PRINT ,,B$
4092 PRINT "TOTAL STOCK=";A(I,1)
+A(I,2)
4095 PRINT ,,"ENTER AMOUNT REMOV
ED FROM STOCK."
4100 INPUT A
4110 IF A<0 THEN GOTO 4100
4111 IF A(I,1)-A<0 THEN PRINT ,,
"NOT POSSIBLE. THAT QUANTITY OF
 STOCK IS NOT YET AVAILABLE."
4112 IF A>A(I,1) AND A(I,2)>0 TH
EN PRINT ,,"AWAITING DELIVERY OF
-"
4113 IF A(I,1)+A(I,2)<A THEN PRI
NT ,,"ORDER ";ABS (A(I,1)+A(I,2)
-A);" OF:-"
4114 PRINT ,,A$(I)
4119 IF A(I,1)-A<0 THEN GOTO 414
0
```

```
4120 LET A(I,1)=A(I,1)-A
4130 PRINT ">> ";A
4140 PRINT ,,"ANY FURTHER STOCK
REMOVED (Y/N)?"
4150 IF INKEY$<>"" THEN GOTO 415
0
4160 IF INKEY$="" THEN GOTO 4160
4170 IF INKEY$="Y" THEN GOTO 400
0
4175 CLS
4180 GOTO 200
5000 CLS
5010 PRINT Z$
5020 INPUT B$
5030 IF B$="" THEN GOTO 5030
5040 FAST
5050 FOR I=1 TO NO
5060 IF A$(I, TO LEN B$)=B$ THEN
 GOTO 5100
5070 NEXT I
5080 SLOW
5090 GOTO 2070
5100 SLOW
5110 PRINT ,,B$
5120 PRINT ,,"ENTER AMOUNT ON OR
DER."
5130 INPUT A
5140 IF A<0 THEN GOTO 5130
5150 PRINT ">> ";A
5160 LET A(I,2)=A(I,2)+A
5170 PRINT ,,"ANY MORE STOCK ON
ORDER (Y/N)?"
5180 IF INKEY$<>"" THEN GOTO 518
0
5190 IF INKEY$="" THEN GOTO 5190
5200 IF INKEY$="Y" THEN GOTO 500
0
5210 CLS
5220 GOTO 200
6000 CLS
6010 PRINT Z$
6020 INPUT B$
6030 IF B$="" THEN GOTO 6020
6040 FAST
6050 FOR I=1 TO NO
6060 IF A$(I, TO LEN B$)=B$ THEN
 GOTO 6100
6070 NEXT I
6080 SLOW
6090 GOTO 2070
6100 PRINT ,,B$
6110 PRINT ,,"ENTER AMOUNT RECEI
VED."
6120 INPUT A
6130 LET A(I,1)=A(I,1)+A
6140 LET A(I,2)=A(I,2)-A
6150 IF A(I,2)<0 THEN LET A(I,2)
=0
6160 PRINT ">> ";A
6170 PRINT ,,"ANY FURTHER STOCK
RECEIVED (Y/N)"
6180 IF INKEY$<>"" THEN GOTO 618
0
6190 IF INKEY$="" THEN GOTO 6190
6200 IF INKEY$="Y" THEN GOTO 600
0
6210 CLS
6220 GOTO 200
7000 FAST
7005 LPRINT "PRODUCTS REQUIRING
RE-ORDERING."
7010 LPRINT "▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄
▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
7015 LET A=0
7020 FOR I=1 TO NO
7030 IF A$(I, TO 6)="      " THE
N GOTO 7500
7040 IF A(I,1)+A(I,2)>A(I,3) THE
N GOTO 7200
7045 LET A=1
7050 LPRINT
7060 LPRINT "PRODUCT:-"
7070 LPRINT A$(I)
7080 LPRINT "SUPPLIER:-"
7090 LPRINT S$(I)
7100 LPRINT "RE-ORDER QTY. ";A(I
,4)
7110 LPRINT "--------------------
-------------"
7200 NEXT I
7500 CLS
7505 IF A=0 THEN LPRINT "NONE."
7510 SLOW
7520 GOTO 200
8000 CLS
8010 PRINT TAB 8;"SAVE STOCK FIL
E.",,"▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄
▄▄▄▄▄▄"
8020 PRINT ,,"ENTER FILE NAME."
8030 INPUT B$
8040 IF B$="" THEN GOTO 8030
8045 PRINT ">> ";B$
8060 PRINT ,,"START TAPE REORDER
 AND PRESS ANYKEY TO SAVE FILE."
8070 IF INKEY$<>"" THEN GOTO 807
0
8080 IF INKEY$="" THEN GOTO 8080
8090 SAVE B$
8100 CLS
8110 PRINT "FILE ";B$
8115 PRINT
8120 GOTO 200
9000 FAST
9005 LPRINT "STOCK/SUPPLIERS LIS
T.",,"▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
9010 FOR I=1 TO NO
9020 IF A$(I, TO 6)="      " THE
N GOTO 9500
9025 LPRINT
9030 LPRINT "STOCK REF. CODE."
9040 LPRINT A$(I)
9050 LPRINT "SUPPLIER."
9060 LPRINT S$(I)
9070 NEXT I
9500 CLS
9505 SLOW
9510 GOTO 200
```

```
              STOCK CONTROL.
OPTIONS:-

   1. ENTER NEW STOCK ITEM.
   2. PRINT ITEM DETAILS.
   3. DELETE OLD STOCK ITEM.
   4. ENTER REMOVED STOCK.
   5. ENTER STOCK ON ORDER.
   6. ENTER STOCK RECEIVED.
   7. PRINT ITEMS TO RE-ORDER.
   8. SAVE STOCK FILE.
   9. PRINT ALL REF. CODES
      AND SUPPLIERS.

CHOOSE OPTION (1-9).

ENTER PRODUCT REFERENCE CODE.
>> FOOD/STOCK C230D

ENTER AMOUNT IN STOCK.
>> 23

ENTER AMOUNT ON ORDER.
>> 43

ENTER RE-ORDER LEVEL.
>> 12

ENTER RE-ORDER QUANTITY.
>> 20

ENTER SUPPLIERS DETAILS.
B. BARTON. LTD.

ANY FURTHER NEW ITEMS (Y OR N)?

PRODUCT DETAILS:-

FOOD/STOCK C230D

PHYSICAL STOCK    :23
QUANTITY ON ORDER:43
TOTAL STOCK       :66
RE-ORDER LEVEL    :12
RE-ORDER QUANTITY:20

SUPPLIER:-
B. BARTON. LTD.

PRESS "C" FOR COPY, ANY OTHER
KEY FOR MENU.

PRODUCTS REQUIRING RE-ORDERING.
```

Example outputs from the program.

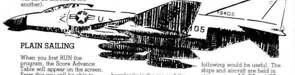
Stephen Ormrod

AIR RAID

Run for cover — and have a smashing time.

In this program, Stephen has taken trouble to incorporate moving graphics, simple rules and an emphasis on strategy rather than relying on Lady Luck.

The rules of the game are quite simple. You begin the game with 30 bombs for your skycraft and you can release a bomb by pressing the '0' key. Your aim is to hit all the enemy ships and planes you can within the restriction of how many bombs you have. If a ship reaches the landing stage before you can destroy it you will lose one of your valuable bombs, and you are further restricted by only being able to fire one bomb at a time (if there is a bomb still on-screen, you will not be allowed to fire another).



PLAIN SAILING

When you first RUN the program, the Score Advance Table will appear on the screen. From this you will be able to see that ships are more valuable (in terms of how many points are scored) than planes. The aircraft are not easy to hit — but there is no penalty for allowing a plane to escape your fire.

After pressing the 'S' key, the score table is erased and you should get your finger over the '0' key — this is the only control key used in the game. Your ammunition will appear as a series of 'grated railings' at the top of the screen. The sea will appear at the bottom of the screen with a landing stage in the bottom right-hand corner. You are positioned in a skycraft, sited just below your ammunition dump at the top of the screen.

Your skycraft slowly 'circles' the sea harbour — you travel across the screen from right to left, and when you disappear on the left-hand side you miraculously appear on the right-hand side again. The white speck in the centre of your skycraft indicates that you have a bomb on the board.

Ships will appear on the left-hand side of the screen and slowly move toward the landing stage. As you move across the screen in your skycraft, you must decide when to release your bomb in order to destroy the ship. If you manage to score a direct hit on the ship, you will be rewarded with the ship being swallowed up by the mysterious deep. However, if you miss the ship, the bomb will explode harmlessly in the sea and the ship will dock at the landing stage, its troops will disembark unopposed and steal one of your bombs.

Periodically, an enemy plane will fly across the screen from left to right, moving twice as quickly as their naval colleagues. They are not trying to reach the landing stage, their object is to get between you and the ships, thus blocking your bomb. There will only ever be one ship or one plane visible on the screen at any time, but a plane and a ship may be seen together.

A SHOT IN THE DARK

One of the problems you will encounter is judging when to release your bombs so that they will destroy the ships. One tactic you could employ is the 'shot in the dark' option in which you release a bomb before a ship has yet come onto the screen. But remember, you have only 30 bombs, so it's best to make them count.

When you have used up your 30 bombs, the display will scroll upwards so that the 'sea' ends up at the top of the screen. The score you have achieved during the game is then displayed, followed by the top highest scores of the day. The sea will then begin rolling again and the ZX81 will wait for you to press '0' signifying that you would like another game.

To help you decipher the program listing, perhaps the following would be useful. The ships and aircraft are held in string arrays, A$, B$ and C$. Cruisers, represented by A$, are generated as follows:

A$(1) — "Space, Graphic 8, Graphic 4, Space"
A$(2) — "Space, Graphic 8, Graphic 5, Space"
A$(3) — "Graphic R, Inverse

Space, Inverse Space, Graphic E"

B$ is used to represent the aircraft and is comprised of the following:

B$ — "Graphic 7, Graphic 7, Graphic 6, Graphic E"

Battleships are held in C$ and are made up as below:

C$(1) — "2 Spaces, Graphic 3,

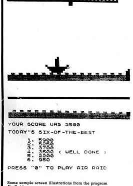
Graphic 5, 2 Spaces"
C$(2) — "2 Spaces, Graphic 8, Graphic 5, 2 Spaces"
C$(3) — "Graphic Y, Graphic 6, Graphic Q, Graphic W, Graphic 6, Graphic T"
C$(4) — "Space, Graphic R, 2 Inverse Spaces, Graphic E, Space"
The ammunition dump is generated in line 134 and comprises 30 Graphic Qs. The string array, D$, is your skycraft and is shown in line 135 as an Inverse Dot and a Space. Line 150 is made up of 31 Graphic As and one Inverse Space.

E$ and F$ are alternating Graphic As and Graphic Ds ending in an Inverse Space. Note that E$ and F$ run in antiphase and alternate in subroutines in lines 7000 and 7020 to make the wave-like motion of the sea.

Lines 200-270 represent the 'core' of the program. The rest of the listing comprises a number of routines which are called from time to time from the main program. Try working out what does what.

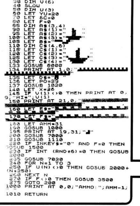
Here is a list of variables used in the program to help you work out how the program works.

Array V(6)— The six best scores.
Array V(3)—Whether or not a plane or ship is visible.
SC— Player's score.
F— Flag controlling whether or not a bomb is loaded aboard your aircraft.
YU— Position of waves on the sea.
PO— Position of last ammunition in the dump.
AMM— The number of bombs left (plus one).
X— The horizontal position of the skycraft.
Y— The horizontal line of the falling bomb.
AS— The controlling variable for what will appear on the screen next.
L— Control variable in loops.
N— Control variable in loops.
SB— Individual score awarded for hitting a ship or plane.



```
YOUR SCORE WAS 3500

TODAY"S SIX-OF-THE-BEST

      1. 5900
      2. 5250
      3. 3700
      4. 3500  < WELL DONE >
      5. 2200
      6. 950

PRESS "0" TO PLAY AIR RAID
```

Some sample screen illustrations from the program Air Raid.



```
AIR RAID
S.A. ORMROD    AUG 1982

  30 DIM V(6)
  40 SLOW
  50 DIM U(3)
  60 LET YU=20
  70 LET SC=0
  80 LET F=0
  85 DIM A$(3,4)
  90 LET A$(1)=" ▐  "
  95 LET A$(2)=" █  "
 100 LET A$(3)="▄██▄"
 105 LET B$="▀▀▄▀"
 110 DIM C$(4,6)
 115 LET C$(1)="  ▐   "
 120 LET C$(2)="  █   "
 125 LET C$(3)="▀▄██▄▀"
 130 LET C$(4)=" ▀██▀ "
 133 GOSUB 6000
 134 PRINT AT 1,0;"▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
 135 LET D$="▪ "
 136 LET PO=29
 137 GOSUB 1020
 140 LET X=26
 145 IF V(1)<>0 THEN PRINT AT 0,
9;"HI=";V(1)
 150 PRINT AT 21,0;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒█"
 160 LET E$="▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒█"
 170 LET F$="▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒▀▒█"
 180 LET AMM=31
 190 GOSUB 1000
 195 PRINT AT 19,31;"▟"
 200 GOSUB 7000
 210 GOSUB 1110
 220 IF INKEY$="0" AND F=0 THEN
GOSUB 1500
 230 IF INT (RND*6)=0 THEN GOSUB
2000
 235 GOSUB 7020
 240 FOR N=1 TO 3
 250 IF U(N)<>0 THEN GOSUB 2000+
(N*250)
 260 NEXT N
 270 IF F<>0 THEN GOSUB 3500
 275 GOTO 200
1000 PRINT AT 0,0;"AMMO:";AMM-1;
" "
1010 RETURN
```

```
1020 PRINT AT 0,20;"SCORE:";SC
1030 RETURN
1110 LET X=X-1
1115 PRINT AT 2,0;CHR$ 0
1120 IF X=-1 THEN LET X=28
1130 PRINT AT 2,X;D$
1140 RETURN
1500 LET F=2
1502 PRINT AT 1,PO;" "
1503 LET PO=PO-1
1505 LET AMM=AMM-1
1507 GOSUB 1000
1510 LET D$="■ "
1515 LET Y=X
1520 RETURN
2000 LET AS=INT (RND*3)+1
2003 IF AS=3 AND U(1)<>0 OR AS=1
 AND U(3)<>0 THEN RETURN
2005 IF U(AS)=0 THEN LET U(AS)=2
2010 RETURN
2250 PRINT AT 17,U(1)-1;CHR$ 0;A
$(1)
2260 PRINT TAB U(1)-1;CHR$ 0;A$(
2)
2270 PRINT TAB U(1)-1;CHR$ 0;A$(
3)
2280 LET U(1)=U(1)+1
2290 IF U(1)<29 THEN RETURN
2300 LET U(1)=0
2310 PRINT AT 17,26;"    "
2320 PRINT TAB 26;"    "
2330 PRINT TAB 26;"    "
2335 GOSUB 3000
2340 RETURN
2500 PRINT AT 10,U(2)-2;"  ";B$
2510 LET U(2)=U(2)+2
2520 IF U(2)<29 THEN RETURN
2530 LET U(2)=0
2540 PRINT AT 10,28;"    "
2550 RETURN
2750 PRINT AT 16,U(3)-1;CHR$ 0;C
$(1)
2760 PRINT TAB U(3)-1;CHR$ 0;C$(
2)
2770 PRINT TAB U(3)-1;CHR$ 0;C$(
3)
2780 PRINT TAB U(3)-1;CHR$ 0;C$(
4)
2790 LET U(3)=U(3)+1
2800 IF U(3)<27 THEN RETURN
2810 LET U(3)=0
2820 PRINT AT 16,26;"      "
2830 FOR N=1 TO 3
2840 PRINT TAB 26;"      "
2850 NEXT N
3000 FOR L=18 TO 1 STEP -1
3002 PRINT AT L,31;"■";AT L+1,31
;" "
3004 NEXT L
3005 FOR L=30 TO PO STEP -1
3006 PRINT AT 1,L;"■ "
3007 NEXT L
3008 LET PO=PO-1
3010 LET AMM=AMM-1
3020 GOSUB 1000
3022 FOR L=PO+1 TO 30
3023 PRINT AT 1,L;" ▟"
3025 NEXT L
3026 PRINT AT 1,31;"▟"
3027 FOR L=2 TO 19
3028 PRINT AT L,31;"▟";AT L-1,31
;" "
3030 NEXT L
3035 IF AMM=1 THEN GOTO 9000
3040 RETURN
3500 LET F=F+1
3505 PRINT AT F,Y;"."
3510 PRINT AT F-1,Y;CHR$ 0
3520 IF F=10 THEN GOTO 4000
3530 IF F<19 THEN RETURN
3540 IF U(1)<>0 THEN GOTO 4500
3550 IF U(3)<>0 THEN GOTO 5000
3555 IF PO=-1 THEN GOTO 9000
3560 IF F=19 THEN RETURN
3570 IF Y>26 THEN PRINT AT 20,Y-
4;"SPLASH"
3580 IF Y<27 THEN PRINT AT 20,Y
"SPLASH"
3590 LET D$="■ "
3595 IF PO=-1 THEN GOTO 9000
3600 LET F=0
3610 RETURN
4000 FOR N=-2 TO 1
4010 IF Y=U(2)+N THEN GOTO 4100
4020 NEXT N
4030 RETURN
4100 FOR N=1 TO 15
4105 PRINT AT 10,U(2)-2;"BOOM"
4110 PRINT AT 10,U(2)-2;"BOOM"
4130 NEXT N
4140 LET SB=INT (RND*4+1)*50
4150 PRINT AT 10,U(2)-2;SB;"    "
4160 LET SC=SC+SB
4170 GOSUB 1020
4180 PRINT AT 10,U(2)-2;"      "
4185 LET U(2)=0
4190 GOTO 3590
4500 FOR N=-1 TO 2
4510 IF Y=U(1)+N THEN GOTO 4600
4520 NEXT N
4530 GOTO 3560
4600 LET U$="     "
4610 PRINT AT 17,U(1)-1;U$
4620 GOSUB 7020
4640 PRINT AT 18,U(1)-1;A$(1)
4650 PRINT TAB U(1)-1;A$(2)
4660 GOSUB 6950
4670 PRINT AT 18,U(1)-1;U$
4680 PRINT TAB U(1)-1;A$(1)
4685 GOSUB 6950
4690 GOSUB 7020
4695 PRINT AT 19,U(1)-1;U$
4700 LET SB=INT (RND*5+1)*200
4705 PRINT AT 19,U(1);SB
4710 LET SC=SC+SB
4715 GOSUB 1020
4720 GOSUB 7500
4730 PRINT AT 19,U(1);
4740 LET U(1)=0
4745 LET F=0
4750 GOTO 3590
5000 FOR N=-1 TO 4
5010 IF Y=U(3)+N THEN GOTO 5100
5020 NEXT N
5030 GOTO 3560
5100 LET U$="      "
5101 PRINT AT 16,U(3)-1;U$
5102 GOSUB 7020
5110 PRINT AT 17,U(3)-1;C$(1)
5115 PRINT TAB U(3)-1;C$(2)
5120 PRINT TAB U(3)-1;C$(3)
5130 GOSUB 6950
5140 PRINT AT 17,U(3)-1;U$
5150 PRINT TAB U(3)-1;C$(1)
5155 PRINT TAB U(3)-1;C$(2)
5157 GOSUB 6950
5160 GOSUB 7020
5165 PRINT AT 18,U(3)-1;U$
5170 PRINT TAB U(3)-1;C$(1)
5175 GOSUB 6950
5180 LET SB=INT (RND*3+1)*100
5185 PRINT AT 19,U(3)-1;U$
5190 PRINT AT 19,U(3);SB
5200 LET SC=SC+SB
5220 GOSUB 1020
5225 GOSUB 7500
5230 PRINT AT 19,U(3);"      "
5235 LET F=0
5240 LET U(3)=0
5250 GOTO 3590
6950 FOR N=1 TO 5
6960 NEXT N
7000 PRINT AT YU,0;E$
7010 RETURN
7020 PRINT AT YU,0;F$
7030 RETURN
7500 FOR N=1 TO 15
7510 GOSUB 7000
7520 GOSUB 7020
7530 NEXT N
7540 RETURN
8000 PRINT TAB 10;"AIR RAID"
8010 PRINT TAB 10;"▀▀▀▀▀▀▀▀▀▀"
8020 PRINT "SCORE ADVANCE TABLE:
"
8030 PRINT
8032 PRINT B$;TAB 10;"50,100,150
 OR 200"
```

```
8034 PRINT TAB 8;"-- ENEMY AIRCR
AFT --"
8036 PRINT
8040 FOR N=1 TO 4
8050 PRINT C$(N)
8060 NEXT N
8070 PRINT AT 8,10;"100,200 OR 3
00"
8080 PRINT AT 9,9;"-- BATTLESHIP
 --"
8085 PRINT AT 12,0;
8090 FOR N=1 TO 3
8100 PRINT A$(N)
8110 NEXT N
8120 PRINT AT 13,10;"200,400 OR
600"
8125 PRINT AT 14,7;"-- TROOPS CA
RRIER --"
8130 PRINT AT 17,6;"PRESS ""S""
TO START"
8140 IF INKEY$<>"S" THEN GOTO 81
40
8150 CLS
8160 RETURN
9000 PRINT AT 5,0;"OUT OF AMMUNI
TION"
9005 FOR L=1 TO 50
9010 NEXT L
9015 FOR L=1 TO 20
9020 SCROLL
9025 NEXT L
9030 PRINT AT 3,0;"YOUR SCORE WA
S ";SC
9040 LET YU=0
9050 FOR N=1 TO 6
9060 IF SC>V(N) THEN GOTO 9300
9070 NEXT N
9080 PRINT AT 5,0;"TODAY""S SIX-
OF-THE-BEST:"
9090 FOR N=1 TO 6
9100 PRINT AT N+6,5;N;".  ";V(N)
9110 NEXT N
9120 PRINT AT 14,0;"PRESS ""0""
TO PLAY AIR RAID"
9130 GOSUB 7000
9140 GOSUB 7020
9150 IF INKEY$<>"0" THEN GOTO 91
30
9160 FAST
9170 CLS
9180 GOTO 40
9300 FOR L=6 TO N+1 STEP -1
9310 LET V(L)=V(L-1)
9320 NEXT L
9330 LET V(N)=SC
9340 PRINT AT N+6,13;"< WELL DON
E >"
9350 GOTO 9080
```

P Greenwood

CODING GRAPHICS

Increase your character size with this great utility.

The purpose of the program is to enable screen displays of large characters or user-defined graphics to be created quickly and easily by calling the character or graphic from a library of symbols stored in an array. On program termination, the screen display is saved in high memory, and is then available for inclusion in a new program.

The program is entertaining to use, and some interesting effects can be achieved by mixing both normal and inverse characters on the screen. The symbol library can hold 54 characters, each designed on a three by five matrix, and as an aid, I enclose a list showing the numeric data required to create 54 possible characters. The program may be used with an incomplete library, any new characters being added as and when required, with the option to re-define any previously created character.

HOW THE PROGRAM WORKS

Lines 20 to 240 are concerned with creating a graphic character, and storing the associated data in an array. The array subscript C is calculated by the subroutine at line 640, after which a check is made at line 100 to see if the character has already been defined. When character definition is completed press Newline to enter the PLOT routine, which works as follows.

Lines 260 and 270 set the initial printing position for the first character to be plotted. Line 290 requests a character, the subroutine at line 640 converting C$ onto the appropriate location C in the array. The individual bytes stored in this string location are then POKEd into the first 15 locations of the line 1 REM statement. This operation is carried out by lines 330 to 390.

Line 410 copies the screen display to above RAMtop. Line 420 puts the chosen character at the initial print position. Lines 430 to 470 scan the keyboard for an input, with any X-Y moves incrementing or decrementing the values held in addresses 16542 and 16544. Line 500 copies the screen display back from above RAMtop, a jump then being made to the 420 where the chosen character is re-printed. Line 470 permits an escape from this loop, enabling the chosen character to be incorporated into the screen display when line 410 is next executed.

When character plotting is completed, pressing Newline enters the invert routine, executed by line 560. Line 570 stores the screen display above RAMtop. The final option is to finish the program by typing Y, or Newline to re-enter the plotting routine.

CREATING THE PROGRAM

Reserve some memory for the machine code routines by typing the following direct commands:

POKE 16388,0
POKE 16389,125
NEW

This sets RAMtop to 32000. Type in the Hex loader program figure 1, ensuring that line 1 contains at least 109 characters. Enter the Hex code as shown in the left-hand column of figure 2, terminating each line of code with Newline. If a mistake is made in typing, pressing Newline will select 'edit' mode indicated by an inverse 'E' being displayed. Now type the correct code and continue, or re-edit as required. Terminate data entry by typing ZZ. Delete all lines except line 1, then press Clear.

As a direct command, type DIM A$(54,5,3) to create an array of 54 characters. Note that this direct command avoids the use of a line number in the program, and so careless use of GOTO cannot accidentally re-DIMension the array and destroy previously entered data. RUN will also destroy data, and so the program should always be restarted using GOTO 1. Add the BASIC lines such that the program is now as shown in figure 3. Type GOTO 1 and use figure 4 to create the characters required, entering a number followed by Newline for each of the 15 entries per character. The character is built up and displayed as each entry is made. When data entry is completed, opt out of this routine by pressing Newline. Step through the options by successful presses of Newline, typing 'Y' in response to 'EXIT PROGRAM?'. To save the program on tape, start the cassette recording, then type GOTO 620.

USING THE PROGRAM

Ensure the RAMtop is set to 32000. The program will start automatically when loaded, but if a re-start is required type GOTO 1. In response to 'CREATE WHICH CHARACTER?' press Newline. The next reply will produce the chosen character in the lower left-hand corner of the screen. Steer the character into place and fix using the keys as prompted.

The next character selected will be initially superimposed on the previous one, and may be manipulated as before. You can opt out of this routine by pressing Newline. For inverse graphics press 'Y' in response to the prompt. A 'Y' response to the next prompt will terminate the program. A screen dump Fig. 5, shows typical characters. The display has now been saved in high memory. To call this display into a new program press New, then enter the BASIC shown in Fig. 6. Run this program, and when the report code 0/60 appears, each program line may be deleted. The graphics may now be displayed, either by a direct command PRINT Z$ or as a program line, eg 10 PRINT Z$ followed by GOTO 10.

A LOAD OF CODE

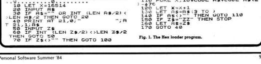
The machine code routines have staring addresses as follows:

16529 C.UP
16541 PRINT
16591 C.DN
16604 INVRT

The routines C.UP and C.DN use the LDIR instruction to block-move 727 bytes of code from the address of the display file to address 32000 and *vice versa*. The Print routine works as follows. The B and A registers hold the X and Y printing offsets with respect to screen 0,0. Steps 19 and 20 increment the display file address by an amount equal to the value in register B. Steps 25 and 26 then increment the display file address by an amount equal to 33 times the value in register A.

Having established the printing position, steps 31 to 34 print the first three bytes of the enlarged character. Steps 35 to 37 then increment the display file address by 30 point to the corresponding printing position on the next line, and the next three bytes are then printed here. This process is repeated five times to print the complete character, at which time the C register, previously loaded with five at step 27, has been incremented to zero and a return to BASIC is made.

The INVRT routine finds each printing character in the display file, adds 128 to the current value, then overwrites the old value.



```
   1 REM .........1.........2...
......3.........4.........5.....
....6.........7.........8.......
..9.........0.........
  10 LET X=16514
  20 INPUT A$
  30 IF A$="" OR INT (LEN A$/2)<
>LEN A$/2 THEN GOTO 20
  40 PRINT AT 21,0;"          ";A
T 21,1;A$
  50 INPUT Z$
  60 IF INT (LEN Z$/2)<>LEN Z$/2
 THEN GOTO 50
  70 IF Z$<>"" THEN GOTO 100
  80 PRINT AT 21,0;"▦"
  90 GOTO 20
 100 SCROLL
 110 POKE X,16*CODE A$+CODE A$(2
)-476
 120 LET X=X+1
 130 LET A$=A$(3 TO )
 140 IF A$<>"" THEN GOTO 110
 150 IF Z$="ZZ" THEN STOP
 160 LET A$=Z$
 170 GOTO 40
```

Fig. 1. The Hex loader program.

```
000000
000000
000000
000000
000000
2A0C40    LD HL,(16396)     C.UP
11007D    LD DE,32000
01D702    LD BC,727
ED80      LDIR
C9        RET
0600      LD B,0            PRINT
3E00      LD A,0
F5        PUSH AF
112100    LD DE,33
2A0C40    LD HL,(16396)
78        LD A,B
FE00      CP 0
2803      JRZ,+3
23        INC HL
10FD      DJNZ,-3
F1        POP AF
FE00      CP 0
2804      JRZ,+4
47        LD B,A
19        ADD HL,DE
10FD      DJNZ,-3
0E05      LD C,5
118140    LD DE,16513
0603      LD B,3
13        INC DE
23        INC HL
1A        LD A,(DE)
77        LD(HL),A
10FA      DJNZ,-6
061E      LD B,30
23        INC HL
10FD      DJNZ,-3
0D        DEC C
20F0      JRNZ,-16
C9        RET
21007D    LD HL,32000       C.DN
ED5B0C40  LD DE,(16396)
01D702    LD BC,727
EDB0      LDIR
C9        RET
2A0C40    LD HL,(16396)     INVRT
0615      LD B,21
0E20      LD C,32
23        INC HL
7E        LD A,(HL)
C680      ADD A,128
77        LD(HL),A
0D        DEC C
20F8      JRNZ,-8
23        INC HL
10F3      DJNZ,-13
C9        RET
```

Fig. 2. The Hex code — load each line of code from the left-hand column and then press Newline.

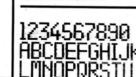


Fig. 3. A screen dump showing typical characters designed using this program.

```
   1 REM                    E£RND) ?
▝NOT ▘ GOSUB ▒TAN ▞ Y( PRINT )5
E£RND? RETURN  C▀7( CLEAR LET RE
TURN  C▖?;( CLEAR :▐)▟RND▖▀▀<7,?(
 IF ▞27( CLEAR $4 LIST TAN 5 ? G
OSUB ?£RND▘NOT  ▘ GOSUB ▒TAN E£RN
D▞+:47 SAVE 7( NEXT TAN
  20 REM ------ ARRAY A$(54,5,3)
                HAS BEEN DIMENSIONED
  30 REM ---------------- CREATE
  40 PRINT AT 21,0;"CREATE WHICH
 CHARACTER?          "
  50 INPUT C$
  60 IF LEN C$>1 THEN GOTO 50
  70 CLS
  80 IF C$="" THEN GOTO 250
  90 GOSUB 640
 100 IF CODE A$(C,1)+CODE A$(C,2
)+CODE A$(C,3)+CODE A$(C,4)+CODE
 A$(C,5)=0 THEN GOTO 140
 110 PRINT AT 21,0;"CHARACTER EX
ISTS, RE-DEFINE? (Y)"
 120 INPUT B$
 130 IF B$<>"Y" THEN GOTO 40
 140 CLS
 150 FOR M=1 TO 5
 160 PRINT """";C$;"""";" LINE "
;M;" DATA?   ";
 170 FOR N=1 TO 3
 180 INPUT D
 190 LET A$(C,M,N)=CHR$ D
 200 PRINT A$(C,M,N);
 210 NEXT N
 220 PRINT
 230 NEXT M
 240 GOTO 40
 250 REM -------------------PLOT
 260 LET X=0
 270 LET Y=16
 280 PRINT AT 21,0;"PLOT WHICH C
HARACTER?  "
 290 INPUT C$
 300 IF C$="" THEN GOTO 520
 310 PRINT AT 21,0;"SEARCHING FO
R CHARACTER"
 320 GOSUB 640
 330 LET Z=16514
 340 FOR M=1 TO 5
 350 FOR N=1 TO 3
 360 POKE Z,CODE A$(C,M,N)
 370 LET Z=Z+1
 380 NEXT N
 390 NEXT M
 400 PRINT AT 21,0;"MOVE 5,6,7,8
, CONFIRM C"
 410 RAND USR 16529
 420 RAND USR 16541
 430 IF INKEY$="5" AND X>0 THEN
LET X=X-1
 440 IF INKEY$="8" AND X<29 THEN
LET X=X+1
 450 IF INKEY$="7" AND Y>0 THEN
LET Y=Y-1
 460 IF INKEY$="6" AND Y<16 THEN
LET Y=Y+1
 470 IF INKEY$="C" THEN GOTO 280
 480 POKE 16542,X
 490 POKE 16544,Y
 500 RAND USR 16591
 510 GOTO 420
 520 REM ---------------- INVERT
 530 PRINT AT 21,0;"INVERSE VIDE
O? (Y)   "
 540 INPUT C$
 550 IF C$<>"Y" THEN GOTO 570
 560 RAND USR 16604
 570 RAND USR 16529
 580 PRINT AT 21,0;"EXIT PROGRAM
? (Y)  "
 590 INPUT C$
 600 IF C$<>"Y" THEN GOTO 280
 610 STOP
 620 SAVE "▒"
 630 GOTO 30
 640 LET C=CODE C$-10
 650 IF C=182 THEN LET C=1
 660 IF C=-10 THEN LET C=54
 670 RETURN
```

Fig. 4. The main part of the BASIC program.

```
" "   0     0     0     0     0     0     0
0     0     0     0     0     0     0     0

""""  5     5     0     0     0     0     0
0     0     0     0     0     0     0     0

"£"   135   3     4     129   4     0     129
4     0     133   0     0     2     3     1

"$"   135   130   0     5     5     1     2
7     1     134   130   1     0     1     0

":"   0     0     0     0     0     0     1
0     0     0     0     0     1     0     0

"?"   6     3     4     1     0     5     0
7     0     0     1     0     0     1     0

"("   135   1     0     5     0     0     5
0     0     134   0     0     0     1     0

")"   134   0     0     0     5     0     0
5     0     135   1     0     1     0     0

">"   0     0     0     134   0     0     135

1     0     1     0     0     0     0     0

"<"   0     0     0     135   1     0     134
0     0     0     1     0     0     0     0

"="   0     0     0     131   131   4     131
131   4     0     0     0     0     0     0

"+"   0     0     0     0     5     0     3
7     1     0     1     0     0     0     0

"-"   0     0     0     0     0     0     3
3     1     0     0     0     0     0     0

"*"   0     4     0     134   130   1     132
128   1     1     5     1     0     0     0

"/"   0     0     5     0     133   0     135
1     0     6     0     0     1     0     0

";"   0     0     0     0     0     0     2
0     0     0     0     0     6     0     0

","   0     0     0     0     0     0     0
0     0     0     0     0     6     0     0

"."   0     0     0     0     0     0     0
0     0     0     0     0     1     0     0

"0"   6     3     4     5     135   5     130
1     5     5     0     5     2     3     0

"1"   132   0     0     133   0     0     133
0     0     133   0     0     3     1     0

"2"   6     3     4     0     135   1     135
1     0     5     0     0     3     3     1

"3"   3     3     5     0     6     0     2
3     4     4     0     5     2     3     0

"4"   0     6     0     6     0     0     5
133   0     3     132   1     0     2     0

"5"   6     3     0     5     0     0     3
3     4     4     0     5     2     3     0

"6"   6     3     4     5     0     0     7
3     4     5     0     5     2     3     0

"7"   3     3     5     0     135   1     135
1     0     133   0     0     2     0     0

"8"   6     3     4     5     0     5     6
3     4     5     0     5     2     3     0

"9"   6     3     4     5     0     5     2
3     5     4     0     5     2     3     0

"A"   6     3     4     5     0     5     7
3     5     5     0     5     1     0     1

"B"   7     3     4     5     0     5     7
3     4     5     0     5     3     3     0

"C"   6     3     4     5     0     0     5
0     0     5     0     4     2     3     0

"D"   7     3     4     5     0     5     5
0     5     5     0     5     3     3     0

"E"   7     3     1     5     0     0     7
3     1     5     0     0     3     3     1

"F"   7     3     1     5     0     0     7
3     1     5     0     0     1     0     0

"G"   6     3     4     5     0     0     5
2     5     5     0     5     2     3     0

"H"   5     0     5     5     0     5     7
3     5     5     0     5     1     0     1

"I"   132   1     0     133   0     0     133
0     0     133   0     0     3     1     0

"J"   0     0     5     0     0     5     0
0     5     5     0     5     2     3     0

"K"   5     0     5     5     6     0     7
4     0     5     2     4     1     0     1

"L"   5     0     0     5     0     0     5
0     0     5     0     0     3     3     1

"M"   7     6     5     5     5     5     5
0     5     5     0     5     1     0     1

"N"   5     0     5     7     4     5     5
2     5     5     0     5     1     0     1

"O"   6     3     4     5     0     5     5
0     5     5     0     5     2     3     0

"P"   7     3     4     5     0     5     7
3     0     5     0     0     1     0     0

"Q"   6     3     4     5     0     5     5
0     5     5     134   1     2     1     1

"R"   7     3     4     5     0     5     7
7     0     5     2     4     1     0     1

"S"   6     3     4     5     0     0     2
3     4     4     0     5     2     3     0

"T"   3     7     1     0     5     0     0
5     0     0     5     0     0     1     0

"U"   5     0     5     5     0     5     5
0     5     5     0     5     2     3     0

"V"   5     0     5     5     0     5     5
0     5     134   135   1     0     1     0

"W"   5     0     5     5     0     5     5
4     5     5     5     5     3     2     1

"X"   5     0     5     2     6     0     0
5     0     6     2     4     1     0     1

"Y"   5     0     5     134   135   1     0
5     0     0     5     0     0     1     0

"Z"   3     3     5     0     135   1     135
1     0     5     0     0     3     3     1
```

Fig. 5. Type in each of the 15 entries you'll need to create each character, followed by Newline.

```
  10 FAST
  20 LET Z$=""
  30 FOR X=32000 TO 32692
  40 IF PEEK X<>118 THEN LET Z$=
Z$+CHR$ PEEK X
  50 NEXT X
  60 SLOW
```

Fig. 6. The BASIC program required to call this display into a new program.

Keith Hewson

SPIRALLING INTO GRAPHICS

The author has developed a fascinating spirograph program which even allows 'impossible' patterns to be generated.

Spirograph patterns are formed by both the interior and exterior epicycloid curves.

There are two basic equations for these curves:

INTERIOR EPICYCLOID:
X=(A-B) COS I+H COS ((A-B) I/B)
Y=(A-B) SIN I — H SIN ((A-B) I/B)
EXTERIOR EPICYCLOID:
X=(A+B) COS I — H COS ((A+B) I/B)
Y=(A+B) SIN I — H SIN ((A+B) I/B)

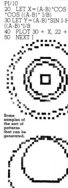
where A is the radius of the large circle (circle A), B is the radius of the small circle (circle B) and H is a point on the circumference of the small circle.

THE EPICYCLOID

This is the curve traced out by the point H on the circumference of the small circle, which rolls on the inside or outside of the large circle. As with the commercial toy, 'Spirograph', the point H can be moved from the circumference to any point along the radius to the centre, where it will just draw a circle.

When the point H is nearer the circumference of the small circle, the patterns tend to be geometric in shape. When point H is moved nearer to the centre of circle B, the curve tends to be more floral, or organic in shape. The general program for the interior epicycloid is:

```
1  INPUT A
2  INPUT B
3  INPUT H
10  FOR I = 0 to 2 *PI STEP
PI/10
20  LET X=(A-B)*COS I+H
*COS ((A-B)* I/B)
30 LET Y=(A-B)*SIN I-H*SIN
((A-B)*I/B
40  PLOT 30 + X, 22 + Y
50  NEXT I
```



Some samples of the sort of patterns that can be generated.

DRAWING PATTERNS

The basic shapes of the hexagon, pentagon, square, triangle and ellipse can easily be entered into this program. Anything higher than a hexagon tends to become indistinct. To draw other patterns, the value for I has to be changed. It varies, on the ZX81, from two to 26 times PI.

There are a few ways of increasing the variety of patterns, but with low resolution graphics, the number is restricted. The three best ways of increasing the variety of patterns is:

1 'Moving to the centre', that is, moving the point H from the circumference of circle B to near its centre. Try this, with A set equal to 20 and B to 10.

```
3  FOR H=2 TO 10 Step 4 60
NEXT H
```

2 'Moving to the side', that is, rotation of the pattern, which can be done by moving an angle, which must be added to the general program. Try it, with A equal 20, B equal 10, H equal to 8, and with the I loop FOR10 to 2 *PI STEP PI/10. You'll need to add the following:

```
4  FOR J=0 TO 3*PI/2
20  LET X=(A-B)*COS
I+HCOS((A-B)*I/B+J)
30  LET Y=(A-B)*SIN
I+H*SIN((A-B)*I/B+J)
60  NEXT J
```

3 'Moving to the centre and the

side', that is, movement of point H and rotation. The program is as in two, with the addition of:

```
3  LET H+0
6  LET H=H + 2
```

There is also the 'extended internal pattern' which is achieved by adding the following to the general program:

```
3  LET F=10
8  FOR H=2 to 8 STEP 2
9  LET F=F40  PLOT F + X,
22 + Y
60  NEXT H
```

RATIOS

Although the low resolution graphics tend to be limiting, the computer offers experimentation in other directions, which are not available on a real Spirograph. Besides the obvious one of a greater variety of ratios, there is the possibility of putting the smaller number in first, and the bigger number in second, as though the larger circle is rolling inside the smaller one.

The following ratios will fit easily into the general program:

A	4	5	6	7	8	9
B	13	13	13	13	13	13
H	10	10	10	10	10	10

And for these use FOR I=O TO 26*PI

A	4	7	8
B	9	9	9
C	8	8	8

For these use FOR I=O TO 18*PI

A	4	6	8
B	11	11	11
H	9	9	9

And for these we suggest FOR I=O TO 22*PI

Some of these have an almost mandala-like quality.

The following can be used in the exterior equations:

A	4	5	6
B	7	7	7
H	6	6	6

And use FOR I=O TO 14*PI

A	4	5
B	9	9
H	6	7

FOR I=O TO 18*PI

EXPERIMENTS

Another direction for experimentation is based on the moving point H, program one. Put the H loop into the I loop. This will draw the points of H first (FOR H=0 TO 6).

And finally, a way of showing the smaller circle moving inside the larger circle, by using the rotation program, is as follows. Put the J loop inside the I loop. This will draw just six circles. Use FOR I=0 to 2*PI STEP PI/10 and FOR J=0 to 2*PI STEP PI/10.

SPECTRUM These super value for money cassettes are only available via MAIL ORDER, from **ORWIN SOFTWARE** **ZX-81**

FOR 16K SPECTRUM (will also run on 48K Spectrum)

CASSETTE A Ten programs for £6

GHOST GOBBLER Eat the ghosts before the ghosts eat you. Eat a star and you can chase the ghosts for a while.

ALIENKILL Control lasers, rockets and a force-field to stop the aliens from landing.
MOUSETRAP Trap the mouse in the corner but not anywhere else.
REVERSI A game of skill with simple rules but sophisticated tactics. Play against the computer.
LASER DEFENCE Control the laser sight to shoot down the alien ships. Machine code sound routines.
TANK BATTLE For 2 players simultaneously or play against the computer. Each player has 2 rotate controls and move and fire controls. The tanks fire steerable missiles.
PHOEBUS A puzzle.
BLACKSPOT Gobble the stars and avoid running into black spots created by crossing your own path.
CUBE Manipulate a cube any size from 2×2×2 to 7×7×7.
MINEFIELD Collect the crowns while avoiding the mines and the electrified fence.
plus an extra . . .
GRAPHIC DEMO Did you know your Spectrum could do all this!?

EDUCATIONAL COMPENDIUM FOR 16K/48K SPECTRUM £6
All these programs on one cassette.

These programs were developed for use in a Junior School and have been in regular use there for some time. Older people also find them interesting and fun.

TORTOISE GRAPHICS: an educational graphics language.
16K version commands include: forward, left and right through a specified angle, wipe, pen up and down, paper and ink colour, macro, edit.
48K version has extra commands including: backward, quit, cancel, printer and tape commands, auto, home, variables, loops.
TORTRAIL & TORTMAZE: 16K games using the forward, left and right commands. These are good for learning a feeling for distances and angles and are fun to play.
REVEAL: an addictive 16K word game. A piece of writing has had its letters rubbed out and you have to find out what they are. You buy letters or try to predict them. Included are 20 textfiles to be solved and a textmaker which will let you input your own text and store it on tape for someone else to solve.
EQUATION INVADERS: a 16K game to make simple equations more interesting.

CENTIPOID PLUS THREE for 48K SPECTRUM. £6

CENTIPOID (machine code).
Centipoid is a full-feature arcade game with mushrooms, spiders, webs, scorpions, gobblers, indestructible dead segments of centipoids that drop down on you, and, of course, the centipoids themselves.
You can choose your own control keys. It is playable from the keyboard or using any joystick interface which, like AGF Interface II, makes the joystick behave like keys on the keyboard.

Plus these games in Basic for the 48K Spectrum on the same cassette.

PANIC HOUND

3D TIC TAC TOE A 3D game of skill played against the computer.

CASSETTE 1: eleven programs (including 7 in machine code): React, Invaders, Phantom Aliens, Maze of Death, Planet Lander, Bouncing Letters, Bug Splat, I Ching, Codebreak, Robots, Basic Hangman.
for 1K ZX-81 £3.80
". . . worth every penny." *Sinclair User, April '84*

CASSETTE 2: ten games in Basic: Reversi, Awari, Laser Bases, Wordmaster, Rectangles, Crash, Roulette, Pontoon, Penny Shoot, Gun Command.
for 16K ZX-81 £5

CASSETTE 3: eight programs (including 1 in machine code): Starship Trojan, Startrek, Princess of Kraal, Battle, Cube, Kalabriasz, Secret Messages, Martian Cricket.
for 16K ZX-81 £5

ZX SCRAMBLE with 3 stages. (machine code) Bomb and shoot your way through the fortified caves.

CASSETTE 4 8 games for 16K ZX-81 £6

GUNFIGHT (machine code)
INVADERS (machine code)
GALAXY INVADERS (machine code) Fleets of swooping and diving alien craft to fight off.
SNAKEBITE (machine code) Eat the snake before it eats you. Variable speed (very fast at top speed).
LIFE (machine code) A ZX-81 version of the well known game.
3D TIC-TAC-TOE (Basic) Played on a 4×4×4 board, this is a game for the brain. It is very hard to beat the computer at it.

". . . well thought-out and the graphics are good for a ZX81 . . . provides value for money." *Sinclair User, April '84*

"quantity as well as quality" *Sinclair User, Oct. '82*

"If each game was on a separate tape and selling for £5 each I would still recommend them" *ZX Computing, Oct/Nov '82*

"Easy to operate, graphically impressive and good value for money." *The Times, 11th Dec '82*

"Definitely good value for money at £6 . . . A smooth clear display and a relatively fast response from the keys. Graphics are as good as you will ever get on the 81." *What Micro? Nov. '83*

7 of the 8 games are in machine code, because this is much faster than Basic.
(Some of these games were previously available form J. Steadman).

CASSETTE 5 8 games for 16K ZX-81 £6

BYTE-MAN (machine code) (previously available from Mindseye)
SPACE RESCUE (machine code) (previously available from Mindseye)
BREAKOUT (machine code)
BLITZ (machine code)
PLANETOIDS (machine code) Rotate, move, fire and hyperspace controls. Wide range of choice of speed and difficulty.
DODGEMS (machine code) Dodge the computer's car while eating the dots.
DRAUGHTS (machine code) Three skill levels.
MERCHANT (Basic) Make your fortune on trading voyages in the Mediterranean and beyond.

"New polish on old favourites . . . the quality of the software and the smooth action displays created on the screen make the programs worthwhile for anyone who has a ZX-81 and plays games using it. . . . It is pleasant to see that Orwin's kind of quality is available again." *Sinclair User, Sept '83*

"Among the best reviewed was Cassette 5 from Orwin Software. For a mere £6, you get eight top quality games . . . All the games are of very high quality and would cost £4 or £5 if sold separately . . . Many of the other software houses could learn a lesson from these." *What Micro? Games supplement, Nov '83*

7 of the 8 games are in machine code because it is much faster than Basic.

Member of the Computer Trade Association

Please make your cheques payable to ORWIN SOFTWARE and send order to ORWIN SOFTWARE, Dept (PER), 25 Brownlow Road, London NW10 9QL.

Prices shown include P&P in UK & Eire. Send SAE for detailed catalogue (state which computer).

Trans cash 5054362 For orders placed at the Post Office

John Wilson

SHOOTER

Simple but fun, that's how we view this game.

Here is a program for the 16K ZX81 which combines machine code with BASIC to form a simple, but fun to play, game.

In this listing you will first need to enter the machine code using a machine code loader program such as the one included in this article. You will need 16K of memory because the machine code looks for 118, the Newline code, which means that unless RAMtop is more than 3.5K the machine code will not allow you to fire at the target.

To enter the machine code, first enter the following program:

1 REM . . . .
2 POKE 16510,0
3 FOR N = 0 to 11
4 INPUT I
5 POKE (16514 + N),I
6 NEXT I

Now, RUN the program and input the following machine code (the commas separating the numbers represent you pressing Newline):

42,14,64,126,254,118,200, 54,22,35,24,247,

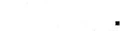
Now, remove all the lines except line 0, and type in the main body of the program shown below.

ON TARGET

Once you have typed in the whole program, type RUN and you are ready to start the game. On the right of the screen, the target will appear and on the left a column of arrows will be seen. When the arrow is in line with the target, press the 'F' key to fire.

You get ten goes to hit the target, after which you will be given your score.

```
  0 REM E:RND7/ RUN
  4 LET S=0
  5 FOR G=1 TO 10
  6 LET L=0
  7 CLS
  6 LET A=INT (RND*15+5)
 10 PRINT AT A,31;"▒"
 11 FOR N=0 TO 25
 12 NEXT N
 20 FOR N=1 TO 21
 30 PRINT AT N,0;">";
 40 IF INKEY$<>"" THEN LET L=US
R 16514
 50 IF L<>0 THEN GOTO 65
 60 NEXT N
 65 IF N=A THEN LET S=S+1
 70 NEXT G
 75 CLS
 80 PRINT "SCORE=";S;" OUT OF 1
0"
 90 PRINT AT 10,0;"PLAY AGAIN
(Y/N)?"
100 IF INKEY$="Y" THEN GOTO 4
110 IF INKEY$="" THEN GOTO 100
```

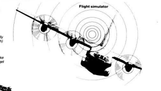


```
>
>
>
>
>
>
>
>
>
>                                ▒
>
>-------------------------------
SCORE=3 OUT OF 10
```

A screen illustration from the program. Shooter.

ORIC & ATMOS 48K Cassettes

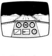
Super 3D arcade action Invaders will test your skill and 3D co-ordination to their limits. 8 rippling colour Invader screens of increasing difficulty with keyboard controlled 3D laser movement and fire sequence. (Brain required) **£6.95**


Flight simulator

Oricle *(Smart Oric)*
Hours of family fun. Oric gets smarter the longer you play, eventually you will have no secrets. Be careful what you say. **£6.95** *Also in French)*

Mystery Tower
Superb musical arrangements. 47 different rooms. 3D and colour make this adventure game most enjoyable. You will certainly get lost, or forget where you're going. **£6.95** *(Also in French)*

Flight Simulator
With airspeed, radar, ground/vertical flight path indicators, artificial horizon and altimeter with continuous flight monitor on engine condition, stall speed etc., full colour, Hi-res graphics, 3D and sound. **£6.95**

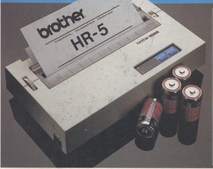


Space Crystal
Space trading adventure game. Fascinating 3D space travel on your voyage in search of the illudium crystal, Black holes, meteorite storms, Space pirates, etc. Full colour and sound. **£6.95.** *Also available for Dragon 32, Spectrum and Commodore 64*

QUARK DATA P.O. BOX 61, Swindon, Wilts. Tel. (0793) 40661.
Please supply. . . . .
. . . .
Prices incl. p & p UK only
Cheque for total amount enclosed . . . .
Name . . . .
Address . . . .
. . . .

Access

Little Brothers should be seen but not heard.



REGULAR, CONDENSED, OR EXTENDED FACES.

CUT SHEET A4 OR ROLLER PAPER.

BATTERY OR MAINS OPERATED.

A maxim which eloquently describes the Brother HR-5.

Less than a foot across, it's nonetheless loaded with features.

But there's one thing the HR-5 won't give you. Earache.

For the annoying 'clickety clack' many printers produce is mercifully absent from the HR-5.

Quietly efficient, it delivers high definition dot matrix text over 80 columns at 30 c.p.s.

The HR-5 also has something of an artistic bent.

Being capable of producing uni-directional graph and chart images together with bi-directional text.

It will also hone down characters into a condensed face, or extend them for added emphasis.

Incorporating either a Centronics parallel or RS-232C interface, the HR-5 is compatible with most home computers and popular software.

Perfectly portable, the battery or mains operated HR-5 weighs less than 4lbs, and has a starting price of only £179.95 (inc.VAT).

Which is really something to shout about.

PLEASE SEND ME MORE DETAILS OF THE REMARKABLE BROTHER HR-5 PRINTER.

NAME____________________

ADDRESS____________________

____

____

__________ TEL.NO.__________ PS/5/84

brother

DEPT P, BROTHER OFFICE EQUIPMENT DIVISION, JONES + BROTHER, SHEPLEY STREET, GUIDE BRIDGE, AUDENSHAW, MANCHESTER M34 5JD
TEL:061 330 6531 (10 LINES) 061 330 0111 (6 LINES) 061 330 3036 (4 LINES). TELEX: 669092 BROTHER INDUSTRIES LTD., NAGOYA, JAPAN.